

## DOCUMENT RESUME

ED 337 637

CE 059 272

AUTHOR Patton, Jan  
TITLE Business Computer Programming I. Curriculum Guide.  
Second Edition.  
INSTITUTION East Texas State Univ., Commerce. Educational  
Development and Training Center.  
SPONS AGENCY Texas Education Agency, Austin.  
PUB DATE 90  
NOTE 80lp.; For volume II, see CE 059 273.  
AVAILABLE FROM Educational Development and Training Center, East  
Texas State University, Commerce, TX 75429 (order no.  
BO-304-CGU).  
PUB TYPE Guides - Classroom Use - Teaching Guides (For  
Teacher) (052)  
EDRS PRICE MF05/PC33 Plus Postage.  
DESCRIPTORS Competency Based Education; \*Computer Science  
Education; Computer Software; Data Processing; High  
Schools; Learning Activities; Mastery Tests; Minimum  
Competency Testing; Office Machines; \*Office  
Occupations Education; \*Programing; State Curriculum  
Guides; Units of Study  
IDENTIFIERS Texas

## ABSTRACT

This guide provides instructors of business computer programming with a curriculum correlated directly to the office education essential elements mandated by the Texas Education Agency. Introductory materials include a scope and sequence and lists of suggested textbooks, resource books, software, audiovisuals, and magazines. Eleven units are divided into these four sections: introduction, computer hardware utilization, program development, and data storage. Each section begins with a list of the specific essential elements. Components of each unit include all or most of the following: (1) the objective page with section and unit titles, essential elements, suggested number of hours, unit objective, student competencies, focus, domains, and basic skills statements; (2) content outline; (3) suggested teacher activities, consisting of general teacher preparation, teacher presentation, reteach activities, and specific teaching activities; (4) handouts; (5) activity and competency test matrices that list which domains and basic skills the activities and competency test parts cover; (6) competency tests; (7) answers to activities and competency tests; and (8) competency checklist. The following student information is provided for most units: unit objective and student competencies, terminology, overview, and activities. Appendixes include reference addresses, course descriptions, office education essential elements, suggested equipment list, lesson plan development, course competency checklist, and suggested modifications for students with special needs. (YLB)

\*\*\*\*\*  
\* Reproductions supplied by EDRS are the best that can be made \*  
\* from the original document. \*  
\*\*\*\*\*

TX90CD10

ED337637

# BUSINESS COMPUTER PROGRAMMING I

## Curriculum Guide

1990  
Edition

U.S. DEPARTMENT OF EDUCATION  
Office of Educational Research and Improvement  
EDUCATIONAL RESOURCES INFORMATION  
CENTER (ERIC)

☒ This document has been reproduced as  
received from the person or organization  
originating it.

☐ Minor changes have been made to improve  
reproduction quality.

• Points of view or opinions stated in this docu-  
ment do not necessarily represent official  
OERI position or policy.

"PERMISSION TO REPRODUCE THIS  
MATERIAL HAS BEEN GRANTED BY

*J.R. Cummings*

TO THE EDUCATIONAL RESOURCES  
INFORMATION CENTER (ERIC)."

2



Educational Development & Training Center  
East Texas State University • Commerce, TX 75429

BO-304-CGU

BEST COPY AVAILABLE

CE059272



# BUSINESS COMPUTER PROGRAMMING I

## CURRICULUM GUIDE

*Prepared by*

*Jan Patton  
Office Education Curriculum Specialist*

*Published by*

*Educational Development and Training Center  
Department of Secondary and Higher Education  
East Texas State University  
Commerce, Texas*

*in cooperation with*

*Texas Education Agency  
Vocational Education Programs  
Austin, Texas*

BO-304-CGU

**SECOND EDITION**  
**1990**

*First printing 1988--300 copies*  
*Second (Revised) Printing 1990--200 Copies*

*All rights reserved. No part of this book may be reproduced in any form  
or by any means without written permission from the publisher.*

*Brand names, company names, trademarks,  
or other identifying symbols appearing in the text are  
used to clarify and identify and do not constitute a  
recommendation or endorsement.*

*To ensure educational equity, this material  
has been reviewed to eliminate  
bias or stereotyping.*



**Educational Development & Training Center**  
East Texas State University • Commerce, TX 75429  
(214)886-5624

## TABLE OF CONTENTS

|                                                                      |          |
|----------------------------------------------------------------------|----------|
| Acknowledgements                                                     | ix       |
| Preface                                                              | xi       |
| <br><b>INTRODUCTORY PAGES</b>                                        |          |
| Organization and Utilization of the Curriculum Guide                 | xiii     |
| Essential Elements for Business Computer Programming                 | xix      |
| Suggested Scope and Sequence                                         | xxi      |
| Resources                                                            | xxiii    |
| <br><b>SECTION I - INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING</b> |          |
| <b>Unit I--Teacher's Guide for Introduction to the Course</b>        | <b>1</b> |
| Suggested Unit Lesson Plan                                           | 5        |
| Content Outline                                                      | 6        |
| Teaching Suggestions                                                 | 7        |
| Handouts                                                             | 11       |
| Transparency Masters                                                 | 15       |
| <b>Unit II--Teacher's Guide for Social Implications</b>              | <b>.</b> |
| Suggested Unit Lesson Plan                                           | 21       |
| Content Outline                                                      | 23       |
| Teaching Suggestions                                                 | 24       |
| Activity and Competency Matrices for Domains and Basic Skills        | 27       |
| Transparency Masters                                                 | 29       |
| Competency Test                                                      | 33       |
| Answers for Student Activities and Competency Test                   | 41       |
| Competency Checklist                                                 | 44       |
| <b>Unit III Teacher's Guide for Computer-Related Concepts</b>        | <b></b>  |
| Suggested Unit Lesson Plan                                           | 45       |
| Content Outline                                                      | 47       |
| Teaching Suggestions                                                 | 49       |
| Activity and Competency Matrices for Domains and Basic Skills        | 51       |
| Handouts                                                             | 53       |
| Transparency Masters                                                 | 63       |
| Supplementary Activity                                               | 91       |
| Competency Test                                                      | 93       |
| Answers for Student Activities and Competency Test                   | 107      |
| Competency Checklist                                                 | 114      |
| <b>Unit I--Student Information for Introduction to the Course</b>    | <b></b>  |
| Unit Objective and Student Competencies                              | 115      |
| Terminology                                                          | 117      |
| Overview                                                             | 119      |
| <b>Unit II--Student Information for Social Implications</b>          | <b></b>  |
| Unit Objective and Student Competencies                              | 125      |
| Terminology                                                          | 127      |
| Overview                                                             | 131      |
| Activities                                                           | 141      |

|                                                                             |         |
|-----------------------------------------------------------------------------|---------|
| <b>Unit III--Student Information for Computer-Related Concepts</b>          |         |
| Unit Objective and Student Competencies                                     | 151     |
| Terminology                                                                 | 153     |
| Overview                                                                    | 157     |
| Activities                                                                  | 171     |
| <br><b>SECTION II - COMPUTER HARDWARE UTILIZATION</b>                       | <br>189 |
| <b>Unit I--Teacher's Guide for Computer Hardware</b>                        |         |
| Suggested Unit Lesson Plan                                                  | 191     |
| Content Outline                                                             | 193     |
| Teaching Suggestions                                                        | 195     |
| Activity and Competency Matrices for Domains and Basic Skills               | 197     |
| Transparency Master                                                         | 199     |
| Supplementary Activity                                                      | 201     |
| Competency Test                                                             | 203     |
| Answers for Student Activities and Competency Test                          | 207     |
| Competency Checklist                                                        | 209     |
| <b>Unit II--Teacher's Guide for Computer Software</b>                       |         |
| Suggested Unit Lesson Plan                                                  | 211     |
| Content Outline                                                             | 213     |
| Teaching Suggestions                                                        | 215     |
| Activity and Competency Matrices for Domains and Basic Skills               | 217     |
| Handouts                                                                    | 219     |
| Supplementary Tests                                                         | 229     |
| Competency Test                                                             | 239     |
| Answers for Student Activities, Supplementary Tests,<br>and Competency Test | 245     |
| Competency Checklist                                                        | 254     |
| <b>Unit III--Teacher's Guide for Computer Operation and Maintenance</b>     |         |
| Suggested Unit Lesson Plan                                                  | 255     |
| Content Outline                                                             | 257     |
| Teaching Suggestions                                                        | 259     |
| Activity and Competency Matrices for Domains and Basic Skills               | 261     |
| Handout                                                                     | 265     |
| Competency Test                                                             | 267     |
| Answers for Student Activities and Competency Test                          | 279     |
| Competency Checklist                                                        | 284     |
| <b>Unit I--Student Information for Computer Hardware</b>                    |         |
| Unit Objective and Student Competencies                                     | 285     |
| Terminology                                                                 | 287     |
| Overview                                                                    | 291     |
| Activities                                                                  | 307     |
| <b>Unit II--Student Information for Computer Software</b>                   |         |
| Unit Objective and Student Competencies                                     | 313     |
| Terminology                                                                 | 315     |
| Overview                                                                    | 321     |
| Activities                                                                  | 339     |

**Unit III--Student Information for Computer Operation  
and Maintenance**

|                                         |     |
|-----------------------------------------|-----|
| Unit Objective and Student Competencies | 345 |
| Terminology                             | 347 |
| Overview                                | 355 |
| Activities                              | 367 |

**SECTION III - PROGRAM DEVELOPMENT** 383

**Unit I--Teacher's Guide for Problem-Solving Concepts and Applications**

|                                                               |     |
|---------------------------------------------------------------|-----|
| Suggested Unit Lesson Plan                                    | 385 |
| Content Outline                                               | 386 |
| Teaching Suggestions                                          | 387 |
| Activity and Competency Matrices for Domains and Basic Skills | 389 |
| Handout                                                       | 391 |
| Competency Test                                               | 393 |
| Answers for Student Activities and Competency Test            | 399 |
| Competency Checklist                                          | 408 |

**Unit II--Teacher's Guide for Programming Concepts and Applications**

|                                                               |     |
|---------------------------------------------------------------|-----|
| Suggested Unit Lesson Plan                                    | 409 |
| Content Outline                                               | 413 |
| Teaching Suggestions                                          | 419 |
| Activity and Competency Matrices for Domains and Basic Skills | 423 |
| Transparency Masters                                          | 429 |
| Competency Test                                               | 439 |
| Answers for Student Activities and Competency Test            | 465 |
| Competency Checklist                                          | 493 |

**Unit I--Student Information for Problem-Solving Concepts  
and Applications**

|                                         |     |
|-----------------------------------------|-----|
| Unit Objective and Student Competencies | 497 |
| Terminology                             | 499 |
| Overview                                | 505 |
| Activities                              | 511 |

**Unit II--Student Information for Programming Concepts  
and Applications**

|                                                        |     |
|--------------------------------------------------------|-----|
| Unit Objective and Student Competencies                | 521 |
| <b>Module A: Setup Procedures</b>                      |     |
| Overview                                               | 525 |
| Activities                                             | 527 |
| <b>Module B: Introduction to Programming</b>           |     |
| Overview                                               | 529 |
| Activities                                             | 533 |
| <b>Module C: Documentation, Debugging, and Testing</b> |     |
| Overview                                               | 539 |
| Activities                                             | 545 |



|                                                               |     |
|---------------------------------------------------------------|-----|
| <b>Module D: Using Arithmetic</b>                             |     |
| Overview                                                      | 549 |
| Activities                                                    | 551 |
| <b>Module E: Input Concepts</b>                               |     |
| Overview                                                      | 557 |
| Activities                                                    | 559 |
| Programs                                                      | 569 |
| <b>Module F: Branching</b>                                    |     |
| Overview                                                      | 573 |
| Activities                                                    | 575 |
| Programs                                                      | 579 |
| <b>Module G: Totals</b>                                       |     |
| Overview                                                      | 583 |
| Activities                                                    | 585 |
| Programs                                                      | 587 |
| <b>Module H: Module Programming</b>                           |     |
| Overview                                                      | 589 |
| Programs                                                      | 591 |
| <b>Module I: Controlled Loops</b>                             |     |
| Overview                                                      | 593 |
| Activities                                                    | 595 |
| Program                                                       | 597 |
| <b>Module J: Advanced Concepts</b>                            |     |
| Overview                                                      | 599 |
| Activities                                                    | 601 |
| Program                                                       | 603 |
| <b>SECTION IV - DATA STORAGE</b>                              | 605 |
| <b>Unit I--Teacher's Guide for Media and Types of Storage</b> |     |
| Suggested Unit Lesson Plan                                    | 607 |
| Content Outline                                               | 608 |
| Teaching Suggestions                                          | 609 |
| Activity and Competency Matrices for Domains and Basic Skills | 611 |
| Transparency Masters                                          | 613 |
| Competency Test                                               | 627 |
| Answers for Student Activities and Competency Test            | 629 |
| Competency Checklist                                          | 631 |
| <b>Unit II--Teacher's Guide for Data Storage Applications</b> |     |
| Suggested Unit Lesson Plan                                    | 633 |
| Content Outline                                               | 634 |
| Teaching Suggestions                                          | 635 |
| Activity and Competency Matrices for Domains and Basic Skills | 637 |
| Transparency Masters                                          | 639 |
| Competency Test                                               | 643 |
| Answers for Student Activities and Competency Test            | 647 |
| Competency Checklist                                          | 654 |

**Unit III--Teacher's Guide for Data Representation  
and Numbering Systems**

|                                                               |     |
|---------------------------------------------------------------|-----|
| Suggested Unit Lesson Plan                                    | 655 |
| Content Outline                                               | 656 |
| Teaching Suggestions                                          | 657 |
| Activity and Competency Matrices for Domains and Basic Skills | 659 |
| Transparency Masters                                          | 661 |
| Competency Test                                               | 685 |
| Answers for Student Activities and Competency Test            | 693 |
| Competency Checklist                                          | 697 |

**Unit I--Student Information for Media and Types of Storage**

|                                         |     |
|-----------------------------------------|-----|
| Unit Objective and Student Competencies | 699 |
| Terminology                             | 701 |
| Overview                                | 705 |
| Activities                              | 707 |

**Unit II--Student Information for Data Storage Applications**

|                                          |     |
|------------------------------------------|-----|
| Unit Objectives and Student Competencies | 711 |
| Terminology                              | 713 |
| Overview                                 | 715 |
| Activities                               | 717 |

**Unit III--Student Information for Data Representation  
and Numbering Systems**

|                                         |     |
|-----------------------------------------|-----|
| Unit Objective and Student Competencies | 723 |
| Terminology                             | 725 |
| Overview                                | 727 |
| Activities                              | 729 |

|                                                 |            |
|-------------------------------------------------|------------|
| <b>CLOSURE OF BUSINESS COMPUTER PROGRAMMING</b> | <b>735</b> |
|-------------------------------------------------|------------|

**APPENDICES**

|                                                  |     |
|--------------------------------------------------|-----|
| Appendix A - Reference Addresses                 | 755 |
| Appendix B - Course Descriptions                 | 775 |
| Appendix C - Office Education Essential Elements | 779 |
| Appendix D - Suggested Equipment List            | 813 |
| Appendix E - Lesson Plan Development             | 815 |
| Appendix F - Course Competency Checklist         | 829 |
| Appendix G - Special Needs                       | 831 |
| Appendix H - 2+2+2 Articulation Project          | 871 |

## ACKNOWLEDGMENTS

Special gratitude and appreciation are extended to the following persons for their cooperation in planning, writing, reviewing, and evaluating the materials developed through this project.

### TEXAS EDUCATION AGENCY

Robert K. Gordon, Office Education Program Director  
Dorothy Baum, Office Education Program Specialist  
David Childs, Office Education Program Specialist

### CONTENT COORDINATORS

Shirley Bray, Coordinator  
Andy Fitch, Assistant

### CURRICULUM COMMITTEE

Sheila Beach, Robert E. Lee High School  
Scott Barrett, Conroe Area Voc. School  
John Steffee, Robert E. Lee High School  
Sondra Ratliff, Trimble Tech. High School

The following Project Advisory Committee members were instrumental in providing a base from which to begin this curriculum guide and contributed in designing the content of this publication:

Scott Barrett, Conroe Area Vocational School  
Sheila Beach, Robert E. Lee High School  
Shirley Bray, Tascosa High School  
Bette Carpenter, Canyon High School

Charles Herold, Milby High School  
Deane Peters, Coppell High School  
Sondra Ratliff, Trimble Tech. High School  
John Steffee, Robert E. Lee High School

This project was also supported by funds from the Carl Perkins Act through a grant sponsored by the Texas Higher Education Coordinating Board. The following individuals provided expertise in development and validation that is evident throughout this curriculum guide.

Dr. Barbara Echord, Project Director  
Dallas County Community College District

Dr. Linda Loomis, Curriculum Consultant

Richard Flemming, Northlake College  
Judy Gammage, Eastfield Community College

Bette Plog, El Centro Community College  
Virginia Salter, Skyline High School

### SECOND EDITION CREDITS

Shirley Bray, Content Consultant

Cheri Halderman, Reviewer  
Allen High School

Transparency masters taken from *Introduction to Data Processing* produced by the Instructional Materials Laboratory at the University of Missouri-Columbia, Columbia, Missouri, July 1981.

Special credit is extended to the support staff of the Educational Development and Training Center for their continuous support and desire to work as a team.

Darlana Murray, Secretary  
Linda Cintron, Student Worker  
Paula Elliott, Editor  
Bob Norsworthy, Graphic Artist  
Linda Baron and Pat Johnson, Word Processing Specialists

Sincere appreciation is extended to these individuals for their time and expert advice in preparing these materials. Special thanks is given to Darlena Murray and Linda Cintron for their diligence in formatting and assembling this curriculum guide.

## PREFACE

The curriculum for Business Computer Programming I was prepared to provide quality instructional materials for instructors of business computer programming courses in Texas. This guide serves as a basis for achieving the office education essential elements mandated by the Texas Education Agency.

This curriculum guide has been developed through a joint effort of the Texas Education Agency and the Educational Development and Training Center located on the campus of East Texas State University in Commerce, Texas.

The purpose of developing this guide is to provide office education instructors with a comprehensive curriculum guide that includes detailed units of instruction correlated directly to the essential elements. Suggestions and comments concerning these curriculum guides are requested and should be sent to the curriculum specialist at the Educational Development and Training Center.

## ORGANIZATION AND UTILIZATION OF THE CURRICULUM GUIDE

Each instructional unit in the office education *Business Computer Programming I* curriculum guide is correlated to the essential elements. The basic components of the curriculum guide are the section pages which are considered the instructor's guide, the unit pages which are considered the student's guide, and the appendices which are a collection of useful, pertinent information. A list of suggested resources is provided in the opening pages.

### SECTION PAGES

Each section may be considered the instructor's guide for the units that are covered within that section. The section provides the instructor with an opportunity to plan and prepare in advance for each unit in the section. The following are the components of the section.

#### Essential Elements

The specific essential elements covered in the section are provided.

#### Suggested Unit Lesson Plan

The suggested unit lesson plan consists of several elements. It is designed to serve as a basis for the preparation of daily and weekly lesson plans and to serve as a basic curriculum outline for each essential element.

**The Objective Page.** Consists of the section and unit titles, essential elements, suggested number of hours, unit objective, student competencies, focus, domains and basic skills statements. It is designed to provide the teacher with data to use in daily and/or weekly lesson plans.

**The Content Outline.** Is a broad outline of the contents of the overview. It is designed to give the teacher a preview of the major emphasis of the unit.

**The Suggested Teacher Activities.** These suggested activities are basic areas that the teacher needs to address for every unit. The suggested teacher activities involve several sections:

- (1) **The General Teacher Preparation** activities suggest general considerations for unit preparation for all units. This section is the same for each unit.
- (2) **The Teacher Presentation** includes setting the stage for the student for optimum classroom learning to take place. This section is the same for each unit.
- (3) **The Reteach Activities** are presented so that they may be used with any unit. This section is the same for each unit.

- (4) The **Specific Teaching Activities** component is extremely important, because it permits the teacher to have materials prepared in advance for the unit being introduced. Specific suggestions may be made as to bulletin boards, equipment needed, guest speakers, field trips, and materials or supplies that are needed. In addition, special instructions are provided for some of the activities and tests in the unit.

### **Activity/Competency Test Matrices for Domains and Basic Skills**

Two matrices are provided for the teacher, listing which domains and basic skills the activities and competency test parts cover. This information may be used for better lesson planning.

### **Competency Tests**

Evaluation is most successful when objectives are clearly stated in advance of learning experiences. The evaluation should be used to measure students' attainment of those specific competencies. Competency tests allow the instructor to make a judgment about student progress in relation to a predetermined standard. These standards should reflect those established by the business community for entry-level employees. It is important that students understand the criteria that are used to evaluate their competency in a given subject area.

Written competency tests based on the unit and student competencies are provided for each unit. These tests are provided to measure the students' understanding of the information covered in the unit overview. The parts of the competency test are queued to the essential elements and student competencies covered in the unit. Individual test items may be used as a pretest measure, as a review, or as an oral quiz to assist the teacher in spotting difficulties being encountered by students in their efforts to accomplish the unit objective. The instructor should construct additional test items for any objectives and information added to the unit. Other information on student evaluation is included in Appendix E--*Steps in Lesson Plan Development*.

The evaluation instruments in this curriculum guide may be used in the following ways:

**Individualized Instruction**--after reading the unit overview or having taken notes from the instructor, the student may complete the activities after brief instructions are provided by the instructor. After successfully completing the activities, the student is then ready for the competency test. If the test results show a weakness in a specific portion of the unit, the student should review that portion of the unit overview, complete reteach activities provided by the instructor, and retake the test.

**Group Instruction**--after participating in classroom discussion and completing the student activities, the student should be able to demonstrate mastery of items listed in the competency checklist for the unit, located in the section. The competency test can be used to measure students' progress toward meeting these competencies.

**Remedial and Reteach Instructions**--students who demonstrate a poor understanding of the subject matter may need remedial help. Selected reteach activities are provided in the Suggested Teacher Activities in the Suggested Unit Lesson Plan.



### Answers to Activities and Competency Tests

Answers are provided for the activities and competency tests included in the unit. Because of the diversified programs, equipment, teaching styles, and student backgrounds, some activities and test questions must be evaluated by the instructor of the course; therefore, no answers are provided.

### Competency Checklists

A Competency Checklist is provided for each unit. The checklist can be used to assist the instructor in recording individual student progress toward the accomplishment of the unit competencies and in recording the mastery of the essential element covered in that specific unit. Space has been provided to record progress for additional reteach activities and other assignments.

The checklists may be accumulated in a file and may serve as a portion of a student's competency profile. A cumulative course competency profile is found in Appendix F.

## UNIT PAGES

The unit pages may be used in a wide variety of ways. Some suggested uses are:

- (1) As a primary resource guide to get started for the beginning instructor.
- (2) As an information base, along with additional textbook information, periodical supplements, and other background experiences.
- (3) As an information base for students that may be reinforced with additional material and lectures.

### Objectives/Competencies

The objective page states the goals of the unit, thus providing a sense of direction and accomplishment for both the student and the teacher. Two types of objectives are provided:

- (1) Unit objectives, which state what the student would achieve upon completion of the unit.
- (2) Student competencies, which state what the student is expected to do to assist himself/ herself in mastering the unit objective.

Each unit of instruction uses the "golden thread" approach. Each student competency is followed through in the unit overview, in the activities, in the competency test, and in the competency profile. Generally, competency one corresponds to activity one and to the first part of the competency test. The "golden thread" approach allows the instructor to easily judge the student's mastery or nonmastery of each competency.

The competencies of the unit provide direction for the learning process; therefore, it is important for the instructor and the students to discuss the unit objective and competencies prior to covering the information in the unit. This discussion will provide a common understanding between the instructor and student of the intent of the competencies.

**It should be noted that when the essential elements were designed, they were only to be the minimum. Instructors should feel free to modify competencies to tailor the material to the needs of the students and the program. When competencies are added, corresponding information, activities, and evaluation should be designed and utilized.**

**Focus.** A focus is provided for each unit. The focus tells "why" the unit should be studied.

**Terminology.** A terminology list including definitions is provided, when appropriate, for most units.

**Unit Overview.** The unit overview is provided as a guide for instructors. It provides information that will specifically address the student competencies and, thus, the essential elements addressed. It should be noted that the overview is not designed as a total source, although it may be used that way. The unit overview is an attempt to condense available current information in order to provide content essential for meeting the cognitive (knowledge) objectives of the unit. The instructor will find that the unit overview serves as an excellent guide for presenting the background information necessary for developing any skills specified in the unit objective. Information may be presented in several ways:

- (1) The instructor may use the unit overview as a guide for presenting the background knowledge necessary for students to complete the specific competencies and, ultimately, the unit objective, or;
- (2) Students may be allowed to use the unit overview as a basic text for a common information base. Information may be added through additional research or lecture notes.

### **Activities**

Learning activities are keyed directly to the student competencies. They provide direction to study and also provide application and reinforcement of cognitive materials presented using the cognitive, psychomotor, and/or affective domains. An effort has been made to incorporate the following elements of the learning process: translation, interpretation, application, analysis, synthesis, and evaluation so that the learning styles of each student will be recognized.

The activity sheets may be given to the students for completion in class or may be used as homework assignments. Answers to the activities are included in the section pages for the individual unit.

The instructor should preview the activities prior to making student assignments to assure that information has been taught and that all required supplies and equipment are available. Specific teaching suggestions are presented in the Suggested Unit Lesson Plan. Class time should be set aside for those activities that require group participation and other small-group work projects. The instructor will need to develop methods by evaluating activities which require student participation.

## **APPENDICES**

A wide variety of material is provided for the instructor in the appendices.

### **Appendix A: Reference Addresses**

Provides the addresses for publishers, software vendors, microcomputer directories, electronic information services, magazines/periodicals/newsletters, services and reports, professional organizations, professional publications, and other business publications.

### **Appendix B: Course Descriptions**

Provides a description of each of the office education courses.

### **Appendix C: Office Education Essential Elements**

Provides a complete set of the office education essential elements.

### **Appendix D: Suggested Equipment List**

Provides a suggested list of equipment and supplies for the course.

### **Appendix E: Steps in Lesson Plan Development**

Provides the teacher with a suggested plan and format for developing lessons. In addition, it provides the teacher with Bloom's Taxonomy and suggested verbs to use with each level to achieve maximum results.

### **Appendix F: Course Competency Checklist**

Provides the teacher with a competency checklist, which includes levels of student achievement, for the entire course.

**Appendix G: Suggested Handicap Modifications**

Provides the teacher with suggested ways of modifying the office education curriculum for handicapped students. It also includes a section on determining students' learning styles.

**Appendix H: 2+2+2 Articulation Project**

Provides the teacher with information on the 2+2+2 program and provides a DACUM chart illustrating what a beginning programmer's responsibilities are.

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**BUSINESS COMPUTER PROGRAMMING I**

(Blended Course)

75.87

(g) **Business Computer Programming (1, 2, or 3 units) shall include the following essential elements:**

- (1) Concepts and skills related to computer data processing. The student shall be provided opportunities to:
  - (A) demonstrate comprehension of computer-related terminology; and
  - (B) identify the major components of computer systems.
- (2) Concepts and skills related to hardware utilization. The student shall be provided opportunities to:
  - (A) review computer hardware operations;
  - (B) understand categorization of hardware according to size, application, and configurations;
  - (C) reinforce the ability to use basic operating systems; and
  - (D) review the proper care of hardware and software.
- (3) Concepts and skills associated with algorithmic approaches to problem solving. The student shall be provided opportunities to:
  - (A) demonstrate the skills related to structured approaches to problem solving; and
  - (B) translate problems into logical steps using appropriate program design techniques.
- (4) Concepts and skills associated with commands and statements of a programming language. The student shall be provided opportunities to:
  - (A) identify and use variables, constants, and literals;
  - (B) demonstrate the use of input and output statements;
  - (C) demonstrate and use conditional and unconditional branching statements;
  - (D) demonstrate and use looping techniques; and

- (E) develop and use subroutines for modular programming.
- (5) Concepts and skills related to program execution, testing, debugging, and documentation. The student shall be provided opportunities to:
- (A) demonstrate the ability to execute computer programs with sample data;
  - (B) distinguish between syntax and logic errors;
  - (C) apply standard debugging procedures; and
  - (D) develop internal and external program documentation including record layout and report spacing chart.
- (6) Concepts and skills associated with data storage. The student shall be provided opportunities to:
- (A) demonstrate an understanding of the types of media for data storage;
  - (B) identify methods of storing data on storage devices; and
  - (C) create and maintain a program to store and retrieve data from a file.
- (7) Concepts and skills associated with the social implications of computers. The student shall be provided opportunities to:
- (A) examine issues concerning responsible use of computer systems; and
  - (B) understand the social ramifications of computer applications related to privacy, values, and reliability of systems.

# SUGGESTED SCOPE AND SEQUENCE

## BUSINESS COMPUTER PROGRAMMING I

| SECTION/UNIT | TITLE                                     | WEEKS | HRS. |
|--------------|-------------------------------------------|-------|------|
| I.           | INTRODUCTION TO BCPI                      |       |      |
| 1.           | Introduction to the Course                | 1/2   | 3    |
| 2.           | Social Implications                       | 1     | 5    |
| 3.           | Computer-Related Concepts                 | 1     | 5    |
| II.          | COMPUTER HARDWARE UTILIZATION             |       |      |
| 1.           | Computer Hardware                         | 1     | 5    |
| 2.           | Computer Software                         | 2 1/2 | 12   |
| 3.           | Computer Operation and Maintenance        | 2 1/2 | 12   |
| III.         | PROGRAM DEVELOPMENT                       |       |      |
| 1.           | Problem-Solving Concepts and Applications | 2     | 10   |
| 2.           | Programming Concepts and Applications     | 11    | 55   |
| IV.          | DATA STORAGE                              |       |      |
| 1.           | Media and Types of Storage                | 1 1/2 | 7    |
| 2.           | Data Storage Applications                 | 1 1/2 | 8    |
| 3.           | Data Representation and Numbering Systems | 1     | 5    |
|              | CLOSURE                                   | 1/2   | 3    |
|              | TOTAL WEEKS/HOURS ALLOCATED:              | 26    | 130  |

Ten weeks have been set aside to cover the common essential elements, review, makeup work, and testing.



## RESOURCES

### SUGGESTED TEXTBOOKS

*ADDISON-WESLEY GUIDE FOR  
ADVANCED PLACEMENT COMPUTER  
SCIENCE*

S. Rhoads

Addison-Wesley Publishing Company

*ADVANCED BASIC: A STRUCTURED  
APPROACH*

Eileen Klimick Schoaff

West Publishers

*ADVANCED STRUCTURED BASIC*

Clark and LaBarre

South-Western Publishing

*ADVANCED STRUCTURED COBOL  
PROGRAM DESIGN AND FILE  
PROCESSING*

Shelly and Cashman

Anaheim Publishing

*ANALYSIS AND DESIGN OF  
INFORMATION SYSTEMS*

James A. Senn

McGraw-Hill Book Company

*ANSI FORTRAN IV--A STRUCTURED  
APPROACH*

William C. Cole

Brown Publishing

*APPLE II INSTANT PASCAL: AN  
INTRODUCTION TO PROGRAMMING*

Moll/Folsom

Houghton-Mifflin Company

*APPLESOFT BASIC FUNDAMENTALS  
AND STYLE*

James S. Quasney and John Maniotes

Boyd and Fraser Publishing Company

*BASIC: A PROBLEM-SOLVING  
APPROACH*

R. Tingey, W. Mott, B. Niebuis

Addison-Wesley Publishing Company

*BASIC CONCEPTS AND STRUCTURED  
PROBLEM SOLVING*

Michel Boillot

West Publishers

*BASIC FOR STUDENTS USING THE  
IBM PC*

M. Trombetta

Addison-Wesley Publishing Company

*BASIC PRINCIPLES OF DATA  
PROCESSING*

James A. Saxon and Wesley W. Steyer

Prentice-Hall, Inc.

*BASIC PROGRAMMING*

Larry Long

Prentice-Hall, Inc.

*BASIC PROGRAMMING TODAY*

Steven L. Mandell

West Publishers

*BASIC PROGRAMMING TODAY: A  
STRUCTURED APPROACH*

Steven L. Mandell

West Publishers

*BASIC: CONCEPTS AND  
STRUCTURED PROBLEM SOLVING*

Michel Boillot

West Publishers

*BIG BLUE BASIC*

Peter Rob

Wadsworth Publishing

*BUSINESS SYSTEM ANALYSIS AND  
DESIGN*

Gary B. Shelly and Thomas J. Cashman

Anaheim Publishing

*COMPLETE BASIC PROGRAMMING*

Steven L. Mandell and Richard M. Wests

Prentice-Hall, Inc.

*COMPLETE BUSINESS BASIC*  
K. Trester, J. Priddy, D. Nielsen  
Addison-Wesley Publishing Company

*COMPUTER ALGORITHMS AND  
FLOWCHARTING*

Gerald A. Silver and Joan B. Silver  
Gregg Community College Division  
McGraw-Hill Book Company

*COMPUTER AND DATA PROCESSING*  
H. L. Capron and Brian K. Williams  
The Benjamin/Cummings Publishing Co.,  
Inc.

*COMPUTER CONCEPTS*  
Marilyn Bohl  
Science Research Associates

*COMPUTER FUNDAMENTALS FOR  
AN INFORMATION AGE*

Gary B. Shelly and Thomas J. Cashman  
Anaheim Publishing

*COMPUTER INFORMATION SYSTEMS*  
David Adams and Gerald Wagner  
South-Western Publishing Company

*COMPUTER PROGRAMMING IN THE  
BASIC LANGUAGE*

French, Jacobs, Moulds, and Schuchman  
Prentice-Hall, Inc.

*COMPUTER PROGRAMMING RGP II*  
Gary B. Shelly and Thomas J. Cashman  
Anaheim Publishing

*COMPUTERS*  
Diane Krasnewich and Timothy N.  
Trainor  
Mitchell Publishing, Inc.

*COMPUTERS*  
Larry Long and Nancy Long  
Prentice-Hall, Inc.

*COMPUTERS AND INFORMATION  
TECHNOLOGY*  
Daggett, Badrkhan, and Kruse  
South-Western Publishing

*COMPUTER AND DATA PROCESSING*  
H. L. Capron and Brian K. Williams  
The Benjamin/Cummings Publishing  
Company

*COMPUTERS AND INFORMATION  
PROCESSING*

William M. Fuori and Lawrence J.  
Aufiero  
Prentice-Hall, Inc.

*COMPUTERS AND INFORMATION  
SYSTEMS*

Jerome S. Burnstein  
CBS College Publishing

*COMPUTERS AND INFORMATION  
SYSTEMS*

Gore and Stubbe  
McGraw-Hill Book Company

*COMPUTERS AND TEACHING*

Gregg Brownell  
West Publishers

*COMPUTERS IN BUSINESS*

Larry Long  
Prentice-Hall, Inc.

*ELEMENTS OF SYSTEMS ANALYSIS,  
4TH EDITION*

Marvin Gore and John Stubbs  
William C. Brown Publishers

*FUNDAMENTALS OF STRUCTURED  
COBOL*

L. Wayne Horn and Gary M. Gleason  
Boyd and Fraser Publishing Company

*GUIDE TO MICROCOMPUTERS*

Fronz J. Frederick  
AECT Publications Sales

*INFORMATION PROCESSING*

Marilyn Bohl  
Science Research Associates

*INTRODUCTION TO COMPUTERS  
AND DATA PROCESSING*

Gary B. Shelly and Thomas J. Cashman  
Anaheim Publishing Company

*INTRODUCTION TO COBOL*  
LR. Overbeek, W. Singletary  
Addison-Wesley Publishing Company

*INTRODUCTION TO COMPUTER  
PROGRAMMING STRUCTURED  
COBOL*  
Gary B. Shelley and Thomas J. Cashman  
Anaheim Publishing

*INTRODUCTION TO COMPUTERS  
AND APPLICATION SOFTWARE*  
Robert H. Blissmer and Robert H. Alden  
Houghton-Mifflin Company

*INTRODUCTION TO COMPUTERS  
AND BASIC PROGRAMMING*  
Kathleen Brenan and Steven L. Mandell  
West Publishers

*INTRODUCTION TO COMPUTERS  
AND INFORMATION PROCESSING*  
Larry Long  
Prentice-Hall, Inc.

*INTRODUCTION TO FLOWCHARTING  
AND COMPUTER PROGRAMMING  
LOGIC*  
Gary B. Shelly and Thomas J. Cashman  
Anaheim Publishing Company

*INTRODUCTION TO PASCAL*  
A. Picciotto  
Addison-Wesley Publishing Company

*JOB CONTROL LANGUAGE*  
Ashley and Fernandez  
Wiley Publishing

*LEARNING COMPUTER  
PROGRAMMING: STRUCTURED  
LOGIC, ALGORITHMS, AND  
FLOWCHARTING*  
Gerald A. Silver and Myrna L. Silver  
Boyd and Fraser Publishing Company

*LOGIC AND STRUCTURED DESIGN  
FOR COMPUTER PROGRAMMERS*  
Harold J. Rood  
PWS Publishing

*MACINTOSH HANDS-ON PASCAL*  
Ted G. Lewis and Abbas Birjandi  
Wadsworth Publishing

*MACINTOSH PASCAL*  
Moll/Folsom  
Houghton-Mifflin Company

*MICROCOMPUTER MANAGEMENT  
AND PROGRAMMING*  
Carol Anne Ogdin  
Prentice-Hall, Inc.

*MICROCOMPUTERS, SOFTWARE,  
AND APPLICATIONS*  
Dennis P. Curtain and Leslie R. Porter  
Prentice-Hall, Inc.

*MINICOMPUTER SYSTEMS*  
M. Vardell Lines and Boeing Computer  
Services  
Winthrop Publishers, Inc.

*MS-DOS ADVANCED APPLICATIONS,  
VOLUME 2*  
David A. Lien and Lewis Rosenfelder  
Compusoft Publishing

*MS-DOS THE BASICS, VOLUME 1*  
David A. Lien and Gary Williams  
Compusoft Publishing

*MVS JCL*  
Wayne Clary  
Murach Publishing

*PASCAL*  
R. Lamb  
Addison-Wesley Publishing Company

*PASCAL: AN INTRODUCTION TO  
METHODICAL PROGRAMMING*  
Findlay/Watt  
Houghton-Mifflin Company

*PASCAL PROGRAMMING TODAY*  
Steven L. Mandell  
West Publishers

*PASCAL--A GENTLE APPROACH*  
Rodney Rambally and Gerard Rambally  
Prentice-Hall, Inc.

*PROBLEM SOLVING AND  
STRUCTURED PROGRAMMING IN  
PASCAL*  
E. Koffman  
Addison-Wesley Publishing Company

*PROBLEM SOLVING WITH  
STRUCTURED FORTRAN 77*  
D.M. Etter  
Addison-Wesley Publishing Company

*PROGRAM DESIGN WITH  
PSEUDOCODE*  
T.E. Bailey and Kris Lundaard  
Cole Publishing Company

*PROGRAMMING BY DESIGN, A FIRST  
COURSE IN STRUCTURED  
PROGRAMMING*  
Philip L. Miller and Lee W. Miller  
Wadsworth Publishing

*PROGRAMMING FOR  
MICROCOMPUTERS APPLE II BASIC*  
Shane  
Houghton-Mifflin Company

*PROGRAMMING LOGIC*  
Leeson  
Science Research Associates

*PROGRAMMING LOGIC FOR  
BUSINESS APPLICATION*  
Pay, Pitter, and Leigh  
Nutchell Publisher

*REASONING WITH A COMPUTER IN  
PASCAL*  
D. Solow  
Addison-Wesley Publishing Company

*STRUCTURED BASIC*  
James F. Clark and William O. Drum  
South-Western Publishing Company

*STRUCTURED BASIC*  
Michael J. Cox and Kathleen Broell  
Sullivan  
Boyd and Fraser Publishing Company

*STRUCTURED BASIC  
FUNDAMENTALS AND STYLE*  
James S. Quasney and John Miniotes  
Boyd and Fraser Publishing Company

*STRUCTURED COBOL  
PROGRAMMING*  
Fenton and Williams  
Times Mirror/Mosby Publishing

*STRUCTURED COBOL--FLOWCHART  
EDITION*  
Gary B. Shelly, Thomas J. Cashman, and  
Steven G. Forsythe  
Anaheim Publishing

*STRUCTURED COBOL--PSEUDOCODE  
EDITION*  
Gary B. Shelly, Thomas J. Cashman, and  
Steven G. Forsythe  
Boyd & Fraser Publishing Company

*STRUCTURED PROGRAMMING  
USING PASCAL*  
Crawley, McArthur, and Miller  
Prentice-Hall, Inc.

*STRUCTURING PROGRAMS IN  
MICROSOFT BASIC*  
Michael J. Cox and Kathleen Broell  
Sullivan  
Boyd and Fraser Publishing Company

*STRUCTURED PROGRAMMING  
LOGIC*  
Thomas W. Mason  
Prentice-Hall, Inc.

*SUCCESSFUL PROBLEM SOLVING  
TECHNIQUES*  
Carole Greens, John Gregory, and Dale  
Seymour  
Creative Publications

**SYSTEMS ANALYSIS AND DESIGN  
METHODS**

Jeffrey Whitten, Lonnie Bentley, and  
Thomas I. M. Ho  
Times Mirror/Mosby College Publishing

**UNDERSTANDING PASCAL: A  
PROBLEM-SOLVING APPROACH**  
Steven L. Mandell and Colleen J. Mandell  
West Publishers

**UNDERSTANDING STRUCTURED  
COBOL, II**  
Michel Boillot  
West Publishers

**USING COMPUTERS IN AN  
INFORMATION AGE**  
Richard W. Grightman and Jeffery M.  
Dimsdale  
Delmar Publishing

**USING MICROPROCESSORS AND  
MICROCOMPUTERS**  
The 6800 Family  
Joseph D. Greenfield, and William C.  
Wray  
Wiley and Sons, Inc.

**WRITING STRUCTURED COBOL**  
D. Johnson  
Addison-Wesley Publishing Company

**SUGGESTED RESOURCE BOOKS**

**ADVANCED 80386 PROGRAMMING  
TECHNIQUES**  
James I. Turley  
Osborne, Borland, McGraw-Hill  
Publishers

**ADVANCED C: SECOND EDITION**  
Herbert Schildt and James I. Turley  
Osborne, Borland, McGraw-Hill  
Publishers

**ADVANCED DBASE III PLUS**  
Miriam Lishin  
Osborne, Borland, McGraw-Hill  
Publishers

**ADVANCED QUICK C**  
Werner Feibel  
Osborne, Borland, McGraw-Hill  
Publishers

**APPLE IIC: USER'S GUIDE**  
Lan Poole  
Osborne, Borland, McGraw-Hill  
Publishers

**APPLE IIGS: TEACHER'S  
REFERENCE**  
Micheal Fischer  
Osborne, Borland, McGraw-Hill  
Publishers

**DOS: THE POCKET REFERENCE**  
Kris James  
Osborne, Borland, McGraw Hill-  
Publishers

**GUIDE TO OS/2**  
Allen Taylor  
Scott, Foresman and Company

**HOW TO RUN A HARD DISK PC**  
Thomas Wrona  
Scott, Foresman, and Company

**MICROS TO MAINFRAMES**  
Ronald F. Kopeck  
Osborne, Borland, McGraw-Hill  
Publishers

**OS/2 MADE EASY**  
Herbert Schildt  
Osborne, Borland, McGraw-Hill  
Publishers

**OS/2: THE POCKET REFERENCE**  
Kris James  
Osborne, Borland, McGraw-Hill  
Publishers

**PC SECRETS: TIPS FOR POWER  
PERFORMANCE**  
James F. Kelly  
Osborne, Borland, McGraw-Hill  
Publishers



***QUICK BASIC: THE COMPLETE REFERENCE***

Steven Nameroff  
Osborne, Borland, McGraw-Hill  
Publishers

***QUICK REFERENCE GUIDE TO MS-DOS COMMANDS***

Van Wolverton  
Microsoft Press

***RUNNING MS-DOS***

Van Wolverton  
Microsoft Press

***SILICON VALLEY: THE MAKING OF THE PC***

Paul Freeberger and Michael Swaine  
Osborne, Borland, McGraw-Hill  
Publishers

***THE COMPLETE APPLE II GS PROGRAMMER'S GUIDE***

Dan Shafer  
Scott, Foresman and Company

***THE MS-DOS ENCYCLOPEDIA***

Microsoft Press

***TURBO C: THE POCKET REFERENCE***

Herbert Schildt  
Osborne, Borland, McGraw-Hill  
Publishers

***TURBO PASCAL VERSION 4.0: THE POCKET REFERENCE***

Kris James  
Osborne, Borland, McGraw-Hill  
Publishers

***USING ROPED FILES***

Ed Jones  
Osborne, Borland, McGraw-Hill  
Publishers

***USING THE MODEL 50 AND 60***

Herbert Schildt  
Osborne, Borland, McGraw-Hill  
Publishers

***USING THE MODEL 50 AND 60***

Mark S. Matthews and Carole Boggs  
Matthews  
Osborne, Borland, McGraw-Hill  
Publishers

***USING TURBO PASCAL VERSION 4.0***

Steven Ward  
Osborne, Borland, McGraw-Hill  
Publishers

***WARD'S PROGRAMMING SECRETS***

Kris James  
Osborne, Borland, McGraw-Hill  
Publishers

***YOUR IBM PC MADE EASY***

Jonathon Sacks  
Osborne, Borland, McGraw-Hill  
Publishers

**SUGGESTED SOFTWARE**

***ACCEL 3 AND 4 COMPILERS FOR TRS BASIC***

DiskCount Data  
TRS-80

***ACE PROGRAMMER***

Instructor's Computer Directory for  
Schools  
Apple II, IIe, IIc 48k  
IBM PC, XT, AT, 128k

***ALCOR PASCAL***

Alcor Systems  
TRS-80

***BASIC***

Abacus Software  
Commodore 64/128

***BASIC INSTRUCTION COURSE***

Also teaches DOS commands  
Alphanetics  
Instructor's Computer Directory  
Instructor Publications  
IBM PC, XT, AT, 128k

**BASIC TEACHER**

AV Systems, Inc.  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k

**BASIC TUTOR SERIES**

Heffernan One-Source  
Apple II, IIe, IIc 48k

**BEGINNING BASIC**

AV Systems, Inc.  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k

**BEGINNING PC BASIC**

Europro, Inc.  
Instructor's Computer Directory  
Instructor Publications  
IBM PC, XT, AT, 128k

**C LANGUAGE**

Alcor Systems  
Easy to use C language compiler  
DiskCount Data  
TRS-80

**COBOL FOR THE TRS 80**

Tandy Corp.  
Instructor's Computer Directory  
Instructor Publications  
TRS-80 model II series

**COMPUTER DISCOVERY FOR JUNIOR  
AND SENIOR HIGH**

Tandy Corp.  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k  
IBM PC, XT, AT 128k

**COMPUTER SCIENCE I, II, III**

CBS Interactive Learning  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k  
IBM PC, XT, AT 128k

**DS TUTOR**

DOS Tutor  
Design Software Inc.  
Learning Services  
IBM PC, XT, AT

**FOR YOUR NEXT ADVENTURE**

Sunburst Communications  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k

**HOW TO PROGRAM IN BASIC  
LANGUAGE TUTORIALS**

Instructor's Computer Directory  
Instructor Publications

**INFORMATION PROCESSING**

*Intro to basic computer functions*  
South-Western Publishing Company  
Apple II, IIe, IIc 48k  
IBM PC, XT, AT 128k

**INTRO SERIES**

Comprehensive Software  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k  
IBM PC, XT, AT 128k

**INTRODUCTION TO PASCAL**

Tandy Corporation  
Instructor's Computer Directory  
Instructor Publications  
TRS-80 model II series

**LEARNING PASCAL SERIES**

J. Weston Walch, Publisher  
Apple II, IIe, IIc 48k

**LEARNING TO USE THE DISK  
OPERATING SYSTEM**

Southern Computer Systems, Inc.  
Workbook & diskette, IBM

**MACHINE LANGUAGE STEP BY STEP**

Heffernan One-Source  
Apple II, IIe, IIc 48k



*MAZE/TURTLE*

EduSoft  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k

*MAZETALK ADVENTURES*

EduSoft  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k

*MICROPASCAL*

J. Weston Walch  
Apple II, IIe, IIc 48k

*MONITOR 5*

DiskCount Data  
TRS-80

*MULTI-BASIC LANGUAGE*

Alcor Systems  
DiskCount Data  
TRS-80

*PASCAL 80*

DiskCount Data  
TRS-80

*PRO-WAM*

DiskCount Data  
TRS-80

*PROFESSOR DOS*

Individual Software, Inc.  
Learning Services

*RAZZLE DAZZLE*

Psychological Software  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k

*SIMULATED COMPUTER*

EduSoft  
Instructor's Computer Directory  
Instructor Publications  
Apple II, IIe, IIc 48k  
TRS-80 model II series  
IBM PC, XT, AT 128k

*SUPER C*

Abacus Software  
Learning Services  
Commodore 64/128

*SUPER PASCAL*

Abacus Software  
Learning Services  
Commodore 64/128

*SUPERIOR BASIC*

DiskCount Data  
TRS-80

*TML PASCAL*

Terrapin  
Learning Services  
Apple IIgs

*TRSCROSS*

DiskCount Data  
TRS-80

*TURBO C*

Borland International  
IBM PC, XT, AT 128k

*TURBO GRAPHIX TOOLBOX*

Borland International  
IBM PC, XT, AT 256k

*TURBO PASCAL*

Borland International  
Apple II, IIe, IIc 48k  
Macintosh, Macintosh +  
IBM PC, XT, AT 128k

*TURBO PASCAL TOOLBOX:*

*NUMERIC METHODS*

Borland international  
Macintosh, Macintosh +  
IBM PC, XT, AT 128k

*TURBO PASCAL TUTOR*

Borland International  
Macintosh, Macintosh +  
IBM PC, XT, AT 128k

*TURBO PROLOG I*

Borland International  
IBM PC, XT, AT 256k

***TURBO PROLOG TOOLBOX***

Borland International  
IBM PC, XT, AT 256k

***TURBO TUTOR***

Borland International  
IBM PC, XT, AT 128k

***ZBASIC***

DiskCount Data  
IBM PC, XT, AT 128k  
Apple II, IIe, IIc 48k  
Kaypro 2, 4, 10 TRS-80

***ZEUS EDITOR/ASSEMBLER***

DiskCount Data  
TRS-80

**SUGGESTED AUDIOVISUALS**

***A COMPUTER GLOSSARY***

16 mm film  
10 min.  
Encyclopedia Britannica Educational  
Corporation

***ADVANCED BASIC TECHNIQUES***

Filmstrip  
Educational Activities, Inc.

***BASIC FOR MICROCOMPUTERS***

Filmstrip  
Educational Activities, Inc.

***CAREERS IN COMPUTING***

Filmstrip  
Educational Activities, Inc.

***COMPUTER APPLICATIONS***

Filmstrip  
Educational Activities, Inc.

***COMPUTER BASICS***

Filmstrip  
Glencoe Publishing

***COMPUTER COMPONENTS***

Vocational Media Associates

***COMPUTER HARDWARE: WHAT IT IS  
AND HOW IT WORKS***

Slides, video, or filmstrip  
The Center for the Humanities

***COMPUTER INDUSTRY***

16 mm film  
11 min.  
AIMS Instructional Media Service, Inc.

***COMPUTER LITERACY AND  
UNDERSTANDING***

Filmstrip  
Educational Activities, Inc.

***COMPUTER PROGRAMMING  
ERRORS: DEBUGGING***

Filmstrip  
Educational Activities, Inc.

***COMPUTER PROGRAMMING: BASIC  
FOR BEGINNING PROGRAMMERS***

Filmstrip  
Educational Activities, Inc.

***COMPUTER VISIONS***

Video  
*An excursion into the world of computers.*  
South-Western Publishing

***COMPUTERS***

16 mm film  
11 min.  
BFA Educational Media

***COMPUTERS AND THE FUTURE***

16 mm film  
30 min.  
Time Life Multimedia

***COMPUTERS, THE FRIENDLY  
INVASION***

16 mm film  
19 min.  
Walt Disney Films

***COMPUTERS, THE INSIDE STORY***

16 mm film  
19 min.  
Churchill Films

*COMPUTER, TOOLS FOR PEOPLE*

16 mm film  
24 min.  
Churchill Films

*COMPUTERS--THE TRUTH OF THE MATTER*

16 mm film  
15 min.  
Walt Disney Films

*COMPUTERS: FROM PEBBLES TO PROGRAMS*

Slides, video or filmstrips

*HANDLING & PROCESSING INFORMATION*

The Center for the Humanities

*HOW THE COMPUTER WORKS*

Filmstrip  
Opportunities for Learning, Inc.

*INTRODUCTION TO BASIC*

Tandy Corporation

*LEARNING DOS AND LEARNING DOS II*

Video  
Career Publishing Incorporated

*PASCAL: A MODERN PROGRAMMING LANGUAGE*

Filmstrip  
Educational Activities, Inc.

*THE COMPUTER*

16 mm film  
11 min.  
AIMS Instructional Media Services, Inc.

*THE COMPUTER POSTER*

Creative Publications

*WORD PROCESSING IN ACTION*

Filmstrip  
Glencoe Publishing

**MAGAZINES**

*ADMINISTRATIVE MANAGEMENT*

P. O. Box 544  
Martinesville, NJ 08836

*APPLE ORCHARD*

908 George Street  
Santa Clara, CA 95050

*BUSINESS SOFTWARE*

M & T Publishing  
P. O. Box 27925  
San Diego, CA 92128

*BYTE: THE SMALL SYSTEMS JOURNAL*

70 Main Street  
Peterborough, NH 03458

*CLASSROOM COMPUTER LEARNING*

Peter Li, Inc.  
2451 East River Rd.  
Dayton, OH 45439

*COMPUTE!*

P. O. Box 5406  
Greensboro, NC 26403

*COMPUTERWORLD*

International Data Corporation  
60 Austin Street  
Newton, MA 02160

*CREATIVE COMPUTING: THE MAGAZINE OF RECREATIONAL AND EDUCATIONAL COMPUTING*

P. O. Box 789-M  
Morristown, NJ 07968

*DATAMATION*

Cahners Publishing Co.  
249 West 17th St.  
New York, NY 10011

*FAMILY COMPUTING*

Scholastic, Inc.  
351 Garver Rd.  
P. O. Box 2700  
Monroe, OH 45050-2700

*INTERFACE AGE:  
MICROCOMPUTING FOR HOME AND  
SMALL BUSINESS*  
13913 Artesia Blvd.  
Cerritos, CA 90701

*KILOBAUD: THE SMALL COMPUTER  
MAGAZINE*  
Peterborough, NH 03458

*PC MAGAZINE*  
39 East Hanover Avenue  
Morris Plains, NJ 07950

*PC WORLD*  
555 DeHaro Street  
San Francisco, CA 94107

*PERSONAL COMPUTING*  
50 Essex Street  
Rochelle Park, NJ 07662

*PERSONAL PUBLISHING*  
Box 390  
Itasca, IL 60143

*POPULAR COMPUTING*  
70 Main Street  
Peterborough, NH 03458

*RUN: THE COMMODORE 64/128  
USER'S GUIDE*  
C. W. Communications  
P. O. Box 954  
Farmingdale, NY 11737

*SOFTWARE DIGEST*  
EDP Services, Inc.  
7620 Turtle Turnpike  
Annadale, VA

*SOFTWARE REVIEW*  
Micorform Review, Inc.  
520 Riverside Drive  
Westport, CT 06880

*THE TECHNOLOGICAL HORIZONS  
IN EDUCATION JOURNAL*  
Information Synergy Inc.  
2626 South Pullman  
Santa Ana, CA 92705

*TEXAS COMPUTING*  
17806 Davenport, Suite 105  
Dallas, TX 75252  
(214) 931-0157

## **SECTION I**

### **INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING**

#### **ESSENTIAL ELEMENT**

**75.87 (g)(1)(A)(B), (7)(A)(B)**

1. Concepts and skills related to computer data processing. The student shall be provided opportunities to:
  - (A) Demonstrate comprehension of computer-related terminology; and
  - (B) Identify the major components of computer systems.
7. Concepts and skills associated with the social implications of computers. The student shall be provided opportunities to:
  - (A) Examine issues concerning responsible use of computer systems; and
  - (B) Understand the social ramifications of computer applications related to privacy, values, and reliability of systems.

## **SECTION I: INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING I**

### **UNIT I: INTRODUCTION TO THE COURSE**

This unit provides an introduction to the course of Business Computer Programming I. It includes an overview of the purpose, objectives, and course contents. Additional subjects such as classroom organization, management, and safety are also included.



## SUGGESTED LESSON PLAN

### SECTION I: INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING I

#### UNIT I: INTRODUCTION TO THE COURSE

**ESSENTIAL  
ELEMENT:**

None

**ALLOCATED**

**TIME: 3 Hours**

**UNIT  
OBJECTIVE:**

Students will develop an understanding of the purpose, contents, and organization of the course Business Computer Programming I.

**STUDENT**

**COMPETENCIES:** After completing this unit, the student will be able to:

1. Relate the purpose of Business Computer Programming I to office education, business, and industry.
2. Recognize the role of the youth organization relative to Business Computer Programming I.
3. Discuss course objectives, equipment, units, competencies, and evaluation measures.
4. Explain classroom organization, management, and safety.

**FOCUS:**

Students will be provided the opportunity to develop and use both academic and vocational skills that will be useful whether they are taking advanced courses or making the transition from high school to the world of work.

## CONTENT OUTLINE

- I. Purpose of course
  - A. Introduction to office education
  - B. Objectives of the course
  - C. Relationship of course to business
- II. Course overview
  - A. Classroom equipment
  - B. Units of instruction
  - C. Competencies
  - D. Evaluation
  - E. Classroom organization and management
  - F. Safety

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

### Specific Teaching Suggestions

The purpose of this unit is to assist you with introducing the course to your students. The introduction is beneficial to both you and the students. It provides you with the opportunity to present any policies, procedures, expectations, and other relative information which enhance classroom instruction and operations. It establishes a working environment for the students and presents an overview for the year's work.

It is suggested that you add other subject matter and instructional materials relative to your teaching assignment. Because the purpose of this section is to introduce the course, student activities and evaluation measures are limited. Some of the teaching activities can be used to develop student activities. It is recommended that some method for assessing student learning be conducted.

1. Design a bulletin board showing the relationship of Business Computer Programming I to office education. A suggestion might be: "Under the Umbrella of Office Education." List cluster courses and specific courses, highlighting Business Computer Programming I.
2. Ask questions at the beginning of the course, such as:
  - a. Why are you taking this course?
  - b. What does the title "Business Computer Programming I" mean to you?
  - c. What do you think the course content covers?
  - d. What is vocational education?
  - e. What is office education?
3. Show and discuss the transparency "Benefits of Office Education."
4. Show and discuss the transparency "Learning through Business Computer Programming I."
5. Ask students to write their perceived likes and dislikes of office occupations. Invite someone who works in an office or related field to respond. Inform the speaker in advance of the nature of the assignment. Secure a person with speaking abilities who projects a positive work image.
6. Post data on job market projections. Update periodically (throughout the year) with projections from different resources.
7. Invite a former office education student to discuss Business Professionals of America. Be sure to inform the speaker in advance what topic you want covered in his/her presentation. You may want to include a question-and-answer period.

**BCPI/Introduction to Business Computer Programming I**  
**Unit 1: Introduction to the Course**  
**Teacher's Guide/TEACHER ACTIVITIES**

8. Invite a former student who was a member of the youth organization to discuss the organization and answer questions.
9. Review youth conferences with students. Competitive event guidelines could be distributed and discussed.
10. Develop an orientation presentation on the office machine and/or equipment in the classroom.
11. Develop, disseminate, and discuss handouts covering classroom policies and procedures. Let students make suggestions for other policies.
12. Disseminate the terminology page. Review the list with the students. Ask that they add to the list any terms which they do not understand. Periodically review the lists with the students. One suggestion is that students identify the term and see if one of the class members knows the meaning.
13. Invite an office worker to speak to the class on the importance of job competency.
14. Show a film relative to one of the topics: work world, fire precautions, or others.

## HANDOUT 1: BUSINESS PROFESSIONALS OF AMERICA

**COLORS:** Navy blue, tan and red

**OFFICIAL TAGLINE:** "Today's students, tomorrow's business professionals"  
Used in all printed materials in conjunction with the name.

**PLEDGE:** We are met in a spirit of friendship and goodwill as we prepare for productive lives in business and office careers. We work together to develop professionalism and leadership through Business Professionals of America and pledge our loyalty to our nation.



**EMBLEM:** The four points of the Business Professionals of America emblem stand for Ambition, Leadership, Sociability and Poise. The shield itself stands for the honor and dignity of the office profession. The quill and inkwell represent the stability of the office through the ages. The stripes represent direction; the torch signifies the ambition of the members as they strive to achieve goals to bring recognition to themselves and to our national organization.

## HANDOUT 2: GOALS OF BUSINESS PROFESSIONALS OF AMERICA

1. **To develop student leadership.** By participating, planning, serving as officers, conducting meetings, and cooperating with fellow students, coordinators, and teachers, students can develop their leadership qualities.
2. **To improve poise, sociability, attitude, and tact.** Members of this organizations improve their social graces by taking an active role in office, classroom, and social activities. Many of the students enrolled in these programs have not previously had the opportunity to participate in an organization. Business Professionals of America gives them the opportunity to acquire a sense of belonging for the first time.
3. **To develop vocational competencies in business occupations.** Vocational competence means being qualified to enter the world of work with a job skill and a certain amount of confidence in one's ability to perform that job. Students in Business Professionals of America are expected to be able to perform at least one job competently and effectively. Of course, their training is more comprehensive than this.
4. **To promote better understanding at local, state, and national levels.** Student participation and fellowship at all levels promotes better programs and provides these students with a knowledge of the people, programs, and activities that are being carried out elsewhere.
5. **To promote student ambition for useful purposes.** Students learn the value of their own personal ambition and derive satisfaction from helping others and from serving.
6. **To learn to plan effectively.** Everyone needs to know where he/she is and where he/she is going. Planning is essential to all successful activities.
7. **To develop an enthusiasm for learning and for remaining knowledgeable in the office held.** In these days, graduating high school seniors can expect to change their major occupation three times. Therefore, enthusiasm for learning is absolutely essential. Students are made aware of our changing society and are taught the necessity of constantly updating their knowledge.
8. **To develop confidence and a spirit of competition.** People are competing during their entire lifetimes. Competitive events create excitement and stimulate and facilitate learning and advancement.
9. **To learn to get along with others.** Even though an individual may have skill and ability, he/she cannot get a job and keep it if unable to get along with fellow workers.
10. **To develop loyalty through esprit de corps.** Everyone needs pride in his or her organization if the organization is to be successful and of benefit to society.
11. **To understand and promote business.** Our "quiet" President, Calvin Coolidge, remarked, "The business of America is business." Our American capitalistic economy makes possible one of the highest standards of living in the world. Business Professionals of America students are proud to be part of this great business world and to be able to help promote it.



# **BENEFITS OF OFFICE EDUCATION**

**DEVELOPMENT OF USEFUL ACADEMIC AND VOCATIONAL  
SKILLS ARE HELPFUL WHEN:**

- (1) TAKING ADVANCED COURSES**
- (2) SEEKING EMPLOYMENT**
- (3) MAINTAINING EMPLOYMENT**

# DEVELOPMENT OF SOCIAL AND LEADERSHIP SKILLS THROUGH PARTICIPATION IN:

## BUSINESS PROFESSIONALS OF AMERICA

- o ATTEND YOUTH CONFERENCES
- o ENTER COMPETITIVE EVENTS
- o RUN FOR AN OFFICE
- o PARTICIPATE IN LOCAL CHAPTER ACTIVITIES

# **LEARNING THROUGH**

## **BUSINESS COMPUTER PROGRAMMING I**

### **INTRO TO BCPI**

INTRO TO COURSE  
SOCIAL IMPLICATIONS  
COMPUTER-RELATED CONCEPTS

19

### **COMPUTER HARDWARE UTILIZATION**

COMPUTER HARDWARE/SOFTWARE  
COMPUTER OPERATION AND  
MAINTENANCE

### **PROGRAM DEVELOPMENT**

PROBLEM-SOLVING CONCEPTS  
AND APPLICATIONS  
PROGRAMMING CONCEPTS AND  
APPLICATIONS

### **DATA STORAGE**

MEDIA AND TYPES OF STORAGE  
DATA STORAGE APPLICATIONS  
DATA REPRESENTATION AND  
NUMBERING SYSTEMS

47

46

## SUGGESTED UNIT LESSON PLAN

### SECTION I: INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING

### UNIT II: SOCIAL IMPLICATIONS

|                               |                               |                            |                |
|-------------------------------|-------------------------------|----------------------------|----------------|
| <b>ESSENTIAL<br/>ELEMENT:</b> | <b>75.87<br/>(g)(7)(A)(B)</b> | <b>ALLOCATED<br/>TIME:</b> | <b>5 Hours</b> |
|-------------------------------|-------------------------------|----------------------------|----------------|

**UNIT  
OBJECTIVE:** Students will describe the issues related to the responsible use of computer systems, appraise the social ramifications of computer usage by society, and identify responsible practices in the use of computers.

**STUDENT  
COMPETENCIES:** After completing this unit, the student will be able to:

1. Identify terminology related to computer crime, computer users, and new computer devices and techniques.
2. Identify the major issues and technological developments related to computers and society.
3. Describe the ethical and legal issues that have arisen as a result of the computer revolution.
4. Identify privacy issues involved with electronic data storage.
5. Explain the reasons for concern about the reliability of computer systems.

**FOCUS:** Using a computer is like using a piece of powerful machinery such as an automobile. Most people use computers harmlessly and responsibly, but if the user is irresponsible, careless, or malicious, the effects of decisions and actions can be far-reaching and damaging. Computers have caused so many changes in the way we live and work that society is totally dependent upon them (as it is on the automobile). The first step toward responsible use of computers is understanding the scope and implications of the relationship between society and technology.

**DOMAINS:** Activities and competency tests have been designed to provide learning experiences utilizing the affective, cognitive, and/or psychomotor domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:** Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests are coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. General issues
  - A. Dependency
  - B. Nameless society
  - C. Technological developments
  - D. Viral epidemic
- II. Legal and ethical issues
  - A. Computer crime
  - B. Data spying
  - C. Hackers
  - D. Employees or former employees
  - E. Piracy
  - F. Laws involving computer and software usage
- III. Privacy issues
  - A. Destination of copies
  - B. Originator of data
  - C. Credit bureaus
  - D. Data banks
  - E. Paperless offices
- IV. Reliability of systems
  - A. Hardware
  - B. Software
  - C. Storage media
  - D. People
  - E. Confidentiality
  - F. Accuracy

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.



Specific Teaching Activities

1. Obtain a copy of the movie *War Games* or another computer-related movie to show in your classroom. The movie *War Games* shows how dangerous "hacking" can be. Discuss with the students the ramifications of computer hacking.
2. Since computer crime and computer ethics have become very important lately, you may wish to have an open discussion before presenting the unit. Using this method, you can decide how much the students already know about the material.
3. The laws that are used in Activity 3, Project 2, were difficult to locate. You may wish to have a guest speaker come to class and give the students an overview of the different laws concerning computer crime and related topics. If you are unable to do this, you can obtain a photocopy or a copy of the *Texas Penal Code* from your local police department. Other laws may be found in history books, publications from the U.S. Government, and periodicals.
4. Prepare a bulletin board that relates to this unit. Ask the students to bring in articles they find in newspapers or periodicals about responsible use of computers and the ramifications that are related to the topic. You may wish to award extra points to the students who participate in this activity.
5. Plan a visit to the school library to search for information about computers and privacy, or legal issues.

### ACTIVITY MATRIX

[illegible]

## COMPETENCY TEST MATRIX

[illegible]

## COMPUTER CRIME

- White-collar crime
- Large amounts of money
- Theft of technology
- Few criminals caught; fewer prosecuted



### PROVISIONS OF THE PRIVACY ACT (1974)

Requires justification for creation of or access to data banks

Requires that government agencies and private industry allow individuals access to information about themselves

## COMPETENCY TEST

### PART I: TERMINOLOGY

**INSTRUCTIONS:** Fill in the blank with the correct term.

- \_\_\_\_\_ 1. A program that mimics the decision-making logic of human experts.
- \_\_\_\_\_ 2. A field of study involving techniques under which computers can formulate responses on the basis of content of accumulated and experience-related information.
- \_\_\_\_\_ 3. Any illegal act for which knowledge of computer technology is used to commit the offense.
- \_\_\_\_\_ 4. The unauthorized copying of a computer program written by someone else.
- \_\_\_\_\_ 5. The science that deals with robots, their construction, capabilities, and applications.
- \_\_\_\_\_ 6. A term describing data that has been translated into a secret code for security reasons.
- \_\_\_\_\_ 7. The final phase of the virus life cycle.
- \_\_\_\_\_ 8. A form of illegal access to data.
- \_\_\_\_\_ 9. An information-producing system aimed at a particular decision a manager must make.
- \_\_\_\_\_ 10. The current compact disks for computers, similar to read-only memory.
- \_\_\_\_\_ 11. The time delay between initial system infection and virus activation.
- \_\_\_\_\_ 12. A way in which the "spirit" of some laws are applied to computer-related activities.
- \_\_\_\_\_ 13. The process of self-isolation within the current host program and attachment of the isolated segment to a new host.

BCPI/Introduction to Business Computer Programming  
Unit 2: Social Implications  
Teacher's Guide/COMPETENCY TEST

- \_\_\_\_\_ 14. Program that watches for typical things viruses do.
- \_\_\_\_\_ 15. A virus that leaves no external signs of its presence after infection.
- \_\_\_\_\_ 16. The process a virus uses to identify and distinguish itself from its host system.
- \_\_\_\_\_ 17. A term that implies that all paper documents have been replaced by electronic media and devices.
- \_\_\_\_\_ 18. The invasion of a virus into a host system.
- \_\_\_\_\_ 19. A computer enthusiast who is challenged by the practice of breaking computer security measures.
- \_\_\_\_\_ 20. A training method that uses a response pad and an interconnection box.
- \_\_\_\_\_ 21. An original virus design.
- \_\_\_\_\_ 22. The remains of a virus after activation.
- \_\_\_\_\_ 23. The program to which the virus attaches itself.
- \_\_\_\_\_ 24. A modification of a viral strain that results in a slightly different functionality.



## PART II: GENERAL ISSUES

### Project 1

**INSTRUCTIONS:** Answer the following questions.

- #### 4) Hypergraphics:

## PART II (Continued)

### Project 2

**INSTRUCTIONS:** Fill in the blank with the correct term.

- \_\_\_\_\_ 1. A segment of machine-language code that replaces software or data with a copy of itself, without authorization.
- \_\_\_\_\_ 2. A virus that checks for a certain action.
- \_\_\_\_\_ 3. A type of virus that was created to spread a greeting of world peace.
- \_\_\_\_\_ 4. A programmer's placement of unauthorized instructions in a computer operating system for future use in gaining access to restricted information.
- \_\_\_\_\_ 5. Replaces the original boot sector with itself; causes loss of data.
- \_\_\_\_\_ 6. A virus that breaks down the directory section, making it impossible for a disk's contents to be read.
- \_\_\_\_\_ 7. Infects any application; loss of data due to system crashes.
- \_\_\_\_\_ 8. A segment of code that allows data to be transmitted to the legitimate operator while other data are transferred to an illegal operator.
- \_\_\_\_\_ 9. Programs that use network connections to spread from system to system.
- \_\_\_\_\_ 10. Replaces original boot sector with itself; spreads quickly to all bootable disks.
- \_\_\_\_\_ 11. A virus that checks for a specific date or time.
- \_\_\_\_\_ 12. Infects any .COM or .EXE program.
- \_\_\_\_\_ 13. Appears in many different varieties.
- \_\_\_\_\_ 14. Infects COMMAND.COM file.

### PART III: LEGAL AND ETHICAL ISSUES

#### Project 1

**INSTRUCTIONS:** Answer each of the following questions.

1. In your own words define computer crime and explain why this type of crime is difficult to prosecute.

2. Name three types of computer crime and discuss what you think should be done to stop the crimes.

1)

2)

3)

### PART III (Continued)

#### Project 2

**INSTRUCTIONS:** Explain in a short paragraph one of the laws that you researched concerning computers.

## PART IV: PRIVACY ISSUES

**INSTRUCTIONS:** List in order of importance those items which you feel could improve privacy for credit bureaus, data banks, and paperless offices.

## PART V: RELIABILITY OF SYSTEMS

**INSTRUCTIONS:** Answer the following questions.

1. List three (3) types of computer hardware.
  - 1)
  - 2)
  - 3)
2. What is the best protection to use against software failure?
3. Name four (4) types of storage media.
  - 1)
  - 2)
  - 3)
  - 4)
4. How can accuracy of data be improved?

## ANSWERS TO ACTIVITIES

### Activity 1

```

COMPUTERETHICS      P
  X                H  A
ENCRYPTION H A      P
  E                O C E
HYPERGRAPHICS K NVIR
  E T            T E L
  P S            PIRACY E
  L V            R S
T I S NETWORKWORMS
I C T            G O
M A E      CDROM  F
E T M            A F
B I              M I
ROBOTICS        V C
M N      TROJANHORSE
B              C
              C
      SCORESVIRUS
              N
      VIRUSREMNANT
    
```

### Activity 2

**Project 1-2** Evaluated to the satisfaction of the instructor.

**Project 3** Answers may vary.

**Robotics:** The science that deals with robots, their construction, capabilities, and applications. This technology can be used as: mail carriers or aides for the handicapped.

**Artificial Intelligence:** A field of study involving techniques under which computers can formulate responses on the basis of content of accumulated and experience-related information. This technology can be used in: decision support systems and expert systems.

**CD-ROM:** The current compact disks for computers, similar to read-only memory; they cannot be erased or recorded on. This technology can be used for archival storage.

**Hyper-graphics:** A method used in classrooms or in-service training that involves the participants, the teacher, and the computer through the use of a response pad and an interconnection box. This technology can be used as a training device.



**Activity 3**

**Projects 1-3** Evaluated to the satisfaction of the instructor.

**Activity 4** Evaluated to the satisfaction of the instructor.

**Activity 5** Answers may vary.

1.
  - 1) Microcomputers
  - 2) Minicomputers
  - 3) Mainframes
  - 4) Keyboards
  - 5) Monitors
  - 6) Printers
2. Back up copies and taking care of the disks.
3.
  - 1) Magnetic tape
  - 2) Diskettes
  - 3) Hard disks
4. Evaluated to the satisfaction of the instructor.

**COMPETENCY TEST ANSWERS**

**Part I**

- |                                 |                      |
|---------------------------------|----------------------|
| 1. Expert systems               | 13. Replication      |
| 2. Artificial intelligence (AI) | 14. Vaccine          |
| 3. Computer crime               | 15. Hidden virus     |
| 4. Piracy                       | 16. Isolation        |
| 5. Robotics                     | 17. Paperless office |
| 6. Encryption                   | 18. Infection        |
| 7. Activation                   | 19. Hacker           |
| 8. Data spying                  | 20. Hypergraphics    |
| 9. Decision support system      | 21. Viral strain     |
| 10. CD-ROM                      | 22. Virus remnant    |
| 11. Activation period           | 23. Host program     |
| 12. Computer ethics             | 24. Viral variety    |

**Part II**

**Project 1** Evaluated to the satisfaction of the instructor.

**Project 2**

- |                   |                           |
|-------------------|---------------------------|
| 1. Computer virus | 8. Trap Door              |
| 2. Logic Bomb     | 9. Network worms          |
| 3. Peace virus    | 10. Pakistani Brain virus |
| 4. Trojan Horse   | 11. Time Bomb             |
| 5. Alameda virus  | 12. Israeli virus         |
| 6. Datacrime      | 13. nVIR                  |
| 7. Scores virus   | 14. Lehigh virus          |

**Part III**

**Project 1-2** Evaluated to the satisfaction of the instructor.

**Part IV** Evaluated to the satisfaction of the instructor.

**Part V** Answers may vary.

1.
  - 1) Monitors
  - 2) Keyboards
  - 3) Printers
2. Make backup copies and protect the disks.
3.
  - 1) Magnetic tape
  - 2) Diskettes
  - 3) Hard disks
  - 4) Disk packs
4. Evaluated to the satisfaction of the instructor.

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(7)(A)(B)

The student shall be provided opportunities to examine issues concerning responsible use of computer systems and understand the social ramifications of computer applications related to privacy, values, and reliability of systems.

**OBJECTIVE:** Students will describe the issues related to the responsible use of computer systems, appraise the social ramifications of computer usage by society, and identify responsible practices in the use of computers.

|                                                                                                             | MASTERY | NON-MASTERY |
|-------------------------------------------------------------------------------------------------------------|---------|-------------|
| 1. Identify terminology related to computer crime, computer users, and new computer devices and techniques. | _____   | _____       |
| 2. Identify the major issues and technological developments related to computers and society.               | _____   | _____       |
| 3. Describe the ethical and legal issues that have arisen as a result of the computer revolution.           | _____   | _____       |
| 4. Identify privacy issues involved with electronic data storage.                                           | _____   | _____       |
| 5. Explain the reasons for concern about the reliability of computer systems.                               | _____   | _____       |
|                                                                                                             | _____   | _____       |

STUDENT NAME \_\_\_\_\_

## SUGGESTED LESSON PLAN

### SECTION I: INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING

#### UNIT III: COMPUTER-RELATED CONCEPTS

|                               |                               |                                    |
|-------------------------------|-------------------------------|------------------------------------|
| <b>ESSENTIAL<br/>ELEMENT:</b> | <b>75.87<br/>(g)(1)(A)(B)</b> | <b>ALLOCATED<br/>TIME: 5 Hours</b> |
|-------------------------------|-------------------------------|------------------------------------|

#### UNIT

**OBJECTIVE:** Students will demonstrate the ability to understand computer processing concepts.

#### STUDENT

**COMPETENCIES:** After completing this unit, the student will be able to:

1. Identify basic computer-related terminology.
2. Outline the functions of data processing.
3. Illustrate the parts of the information processing cycle and explain how each part is interdependent within the cycle.
4. Illustrate parts of the computer processor.
5. Compare the similarities and differences among various computer systems.
6. Outline the events in history pertinent to breakthroughs in data processing.
7. Describe and categorize characteristics of computer languages.

#### FOCUS:

Knowledge of general computer operation and background is necessary in order to understand the more complex computer programs.

1. The ability to identify basic computer-related terms is essential in the day-to-day use of computer-related concepts.
2. Understanding each part of the information processing cycle and how the parts work together is imperative. The student can then process data and, at the same time, understand how the data were processed.

BCPI/Introduction to Business Computer Programming  
Unit 3: Computer-Related Concepts  
Teacher's Guide/LESSON PLAN

3. Identification of various computer systems is important in choosing the system most suitable for the task at hand.
4. Understanding the history of data processing is necessary to understand the development of computers.
5. Categorization and definition of computer languages allows the student to decide which language is most effective in specific problem solving.

**DOMAINS:**

Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:**

Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. Definition of data processing
- II. Functions of data processing
  - A. Data collecting
  - B. Recording
  - C. Sorting
  - D. Classifying
  - E. Calculating
  - F. Storing and retrieving
  - G. Summarizing
  - H. Communicating
- III. Information processing cycle
  - A. Input
  - B. Processing
  - C. Output
  - D. Auxiliary storage
- IV. Types of computers
  - A. Supercomputers
  - B. Mainframe
  - C. Minicomputer
  - D. Microcomputer
  - E. Laptops
- V. History of data processing
  - A. Ancient times to 1800
  - B. Devices and contributions 1800-1900

**BCPI/Introduction to Business Computer Programming**  
**Unit 3: Computer-Related Concepts**  
**Teacher's Guide/CONTENT OUTLINE**

- C. Devices and contributions 1900-present
- D. Computer generations
- VI. Computer languages
  - A. Categories of languages
  - B. Machine language
  - C. Assembly language
  - D. High-level language
  - E. Selecting computer languages
  - F. Language translators



## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

Specific Teaching Activities

1. Bulletin board ideas:

Data processing concepts  
History of data processing  
Information processing cycle  
Types of computers  
Computer languages

2. Use transparencies as needed.

3. Secure samples of:

|                                                  |                                                                     |
|--------------------------------------------------|---------------------------------------------------------------------|
| old media:                                       | floppy diskette<br>magnetic tape<br>punched paper tape<br>disk pack |
| old hardware:                                    | vacuum tube<br>transistors<br>chips<br>wafers                       |
| samples of programs written in various languages |                                                                     |

### ACTIVITY MATRIX

| Activities  | Domains   |              |           | Basic Skills |         |      |
|-------------|-----------|--------------|-----------|--------------|---------|------|
|             | Cognitive | Psycho-motor | Affective | Reading      | Writing | Math |
| 1           | X         |              |           | X            |         |      |
| 2           | X         |              |           |              | X       |      |
| 3 Project 1 | X         | X            |           |              | X       |      |
| Project 2   | X         | X            |           |              | X       |      |
| Project 3   | X         | X            |           |              | X       |      |
| Project 4   | X         | X            |           | X            | X       |      |
| 4 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         |              |           | X            |         |      |
| Project 3   | X         |              | X         | X            | X       |      |
| Project 4   | X         | X            |           | X            | X       |      |
| 5 Project 1 | X         |              |           | X            |         |      |
| Project 2   | X         | X            |           |              | X       |      |
| Project 3   | X         |              |           |              | X       |      |
| Project 4   | X         |              |           |              | X       |      |
| 6 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         |              |           | X            | X       |      |
| Project 3   | X         |              |           |              | X       |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |

[illegible]

## HANDOUT 1

### PC COMPARISON CHART

|                                            | PC XT                    | PC AT                     | 386                                           |
|--------------------------------------------|--------------------------|---------------------------|-----------------------------------------------|
| USES                                       | Personal/Business        | Personal/Business         | Personal/Business<br>Graphics/CAD/<br>Network |
| MICROPROCESSOR                             | Intel 8088               | Intel 80286               | Intel 80386                                   |
| DATAWORD                                   | 8 bits                   | 16 bits                   | 32 bits                                       |
| SPEED                                      | 4- 8MHz                  | 6-12 MHz                  | 12-20 MHz                                     |
| OPERATING SYSTEM                           | MS-DOS                   | MS-DOS-OS/2               | MS-DOS-OS/2                                   |
| ADDRESSABLE<br>MEMORY                      | 640 K                    | 16Mb                      | 4Gb                                           |
| DISK STORAGE<br>5.25 floppy<br>3.5 stiffer | 360K DS/DD<br>720K DS/DD | 1.2M DS/QD<br>1.44M DS/QD | 1.2M DS/QD<br>1.44M DS/QD                     |
| COST                                       | \$400-\$2000             | \$800-\$2000              | \$1700-\$8000                                 |
| STANDARD                                   | 1983-1986                | 1986-1987                 | 1988-                                         |
| USERS/<br>PROGRAMS                         | 1/1                      | 1/MANY                    | MANY/MANY                                     |

## HANDOUT 2

### DEVELOPMENT OF THE IBM PERSONAL COMPUTER

Company strategists never expected the PC to end up on corporate desktops. They set up a maverick outfit, the Entry Systems Division, and permitted its staff to break all unwritten corporate rules. They built the PC with off-the-shelf components from other companies. IBM then sold the PC in computer stores and did not try to make it compatible with everything else on the market! Critics argued that it was no big deal. All it could do was run CP/M programs that had been ported over from 8-bit machines. In fact, that was true. There was no software for the new machine. Some of the PC's early programs included WordStar, dBASE, and SuperCalc, all of which had migrated from the CP/M world. IBM saw the machine appealing to the same crowd buying the Apple II. IBM expected sales to top out at 250,000 units, a figure so meager that some of IBM's marketers refused to handle such a puny account! IBM eventually sold over 3 million units.

- Announced 8/12/81
- 16K or 64K, expandable to 544K
- Two 160K disk drives
- 8088 microprocessor, 16-bit, but with 8-bit bus to be compatible with current I/O devices
- PC-DOS 1.0
- PC-DOS 1.1 for 320K drives
- PC-DOS 2.0 for 360K drives

IBM's impact on personal computing has been very positive. It raised the stakes of the personal computer business substantially when it introduced the IBM PC. The IBM PC used a 16-bit processor when every other microcomputer had an 8-bit processor. The IBM PC offered more memory, making sophisticated business software possible. This in turn provided a platform for programs that revolutionized the personal computer industry and made it grow rapidly.

### IBM PC XT

Announced 8/8/83  
256K on motherboard  
10MB hard disk  
PC-DOS 2.0 for hard disk  
\$4995  
Sold over 2 million units

### IBM PCjr

Announced 11/1/83  
64K  
Color monitor  
Optional 360K disk drive  
PC-DOS 2.1 for 1/2 height disk drive  
Keyboard that interfaced with the system unit via an infrared beam. To ensure that the PCjr did not encroach on its big brother's sales, IBM purposely gave the machine only an approximation of PC compatibility. Home users were put off by the \$699 price tag. IBM came to the rescue with a multimillion-dollar ad campaign, 128K, a second disk drive and a redesigned keyboard. Sold an estimated 530,000 units; April 1985, production stopped.

### IBM Portable PC

Announced 2/16/84  
256K  
Did not support hard disks, lacked a parallel port; IBM's tardy and half-hearted response to the Compaq portable. IBM made the mistake of assuming that the Compaq was successful merely because it was portable, not because IBM did not have enough PC's to meet user demand. They crippled the Portable intentionally so it would not impact PC sales. The machine limped along for 16 months before IBM put it out of its misery.

### IBM PC AT "Advanced Technology"

Announced 8/14/84  
80286 16-bit microprocessor  
6 MHz  
20MB hard disk  
1.4MB floppy disk  
Problems with the hard disk disillusioned many users. Clone makers brought out bigger, swifter, and cheaper 80286 machines. Estimated 650,000 units sold.



### IBM PC II

Scaled down AT with an 80186 microprocessor  
July 1985, project cancelled  
(Tandy 2000 used the 80186 and was likewise scrapped after a limited production cycle.)

### IBM PC AT Enhanced

8 MHz  
New style keyboard  
Pulled into market well behind faster peers.

### IBM PC Convertible

Announced 4/2/86  
IBM's tardy entry into the laptop wars is a solid, reasonably priced machine that has failed to inspire the masses.

### IBM PC XT/286

Announced 9/2/86  
Expensive, incongruous hybrid that mated the AT's motherboard with the XT's chassis, and which is too small to hold AT boards. (Tandy 1000TX is an XT compatible using the 80286 microprocessor but without the capabilities of the AT.)

### IBM PC/386SX

80386 processor  
16-bit bus width  
Speed runs between 286 and 386.

### IBM PC/386

80386 processor  
Memory-management unit makes it more sophisticated than 286. Allows for multiprocessing of programs.  
Clock speed is faster--will process two times faster than the 286.  
Runs at 26.6 SI rating.  
32-bit bus width

### IBM 486

80486 processor  
Announced May 1989. Same as 386 but has a built-in math co-processor.

### IBM PS/2 family

Announced 4/2/87

#### Model 25

Announced 8/1/87  
8086 microprocessor  
8 MHz  
720K 3.5-inch disk  
PC-DOS 3.3 for 3.5-inch drives  
Combined system unit, disk drives and monitor, cannot run the new OS/2 operating system, has Vector Graphics Array monitor.

#### Model 30

8086 microprocessor  
8 MHz  
720K disk drive; 20MB hard drive  
640K  
PC-DOS 3.3, cannot run the new OS/2 operating system, has Vector Graphics Array monitor.

#### Model 50

80286 microprocessor  
10 MHz  
1.44MB 3.5-inch disk; 20MB hard disk  
PC-DOS 3.3 and OS/2 for multitasking  
1MB on motherboard  
3 expansion slots

#### Model 60

80286 microprocessor  
10 MHz  
1.44MB 3.5-inch disk; 44 or 70MB hard disk  
PC-DOS 3.3 and OS/2 for multitasking  
1MB expandable to 15MB  
7 expansion slots

**Model 80**

80386 microprocessor  
16 MHz  
1.44MB 3.5-inch disk; 44, 70, 115MB hard  
disk  
PC-DOS 3.3 and OS/2 for multitasking  
2MB expandable to 16MB  
7 expansion slots

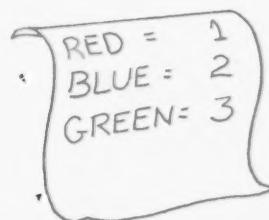
**Laptop**

Vary in size, shape, and power. Growing  
smaller and more powerful with each  
announcement. Generally less than nine  
pounds.

## DATA PROCESSING OPERATIONS



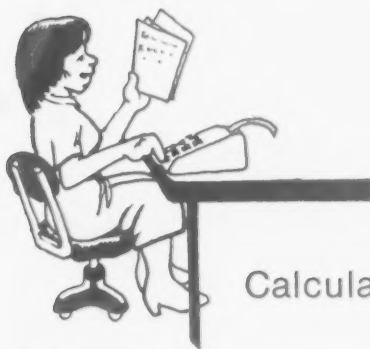
Recording



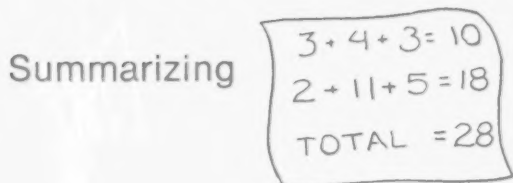
Classifying (coding)



Sorting



Calculating



Summarizing



Storing



Communicating

## WHAT IS DATA PROCESSING?

Data (information)

$$\begin{array}{r} 2 \\ + 2 \\ \hline \end{array}$$

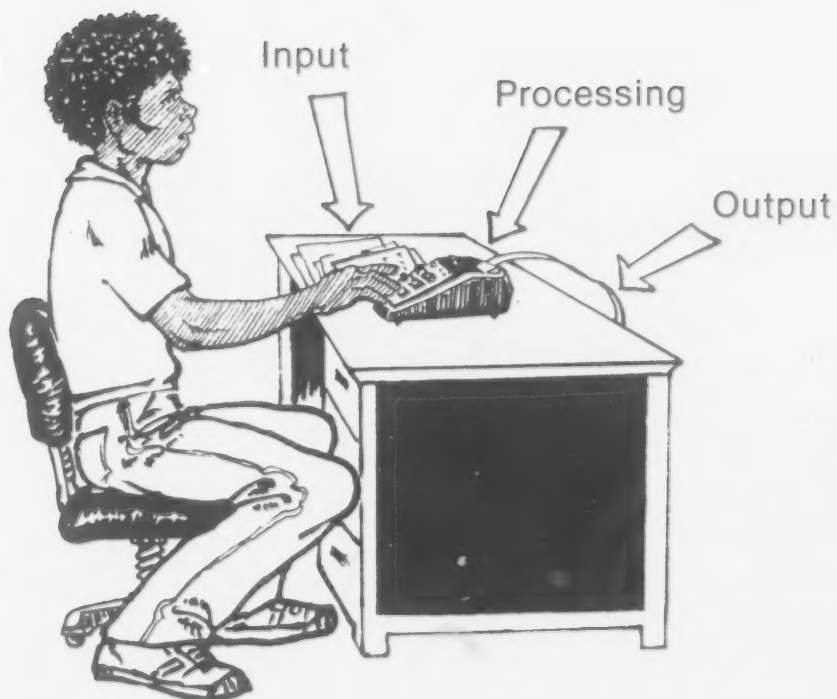
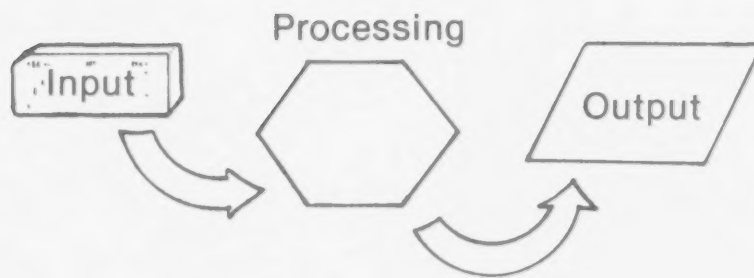
Processing



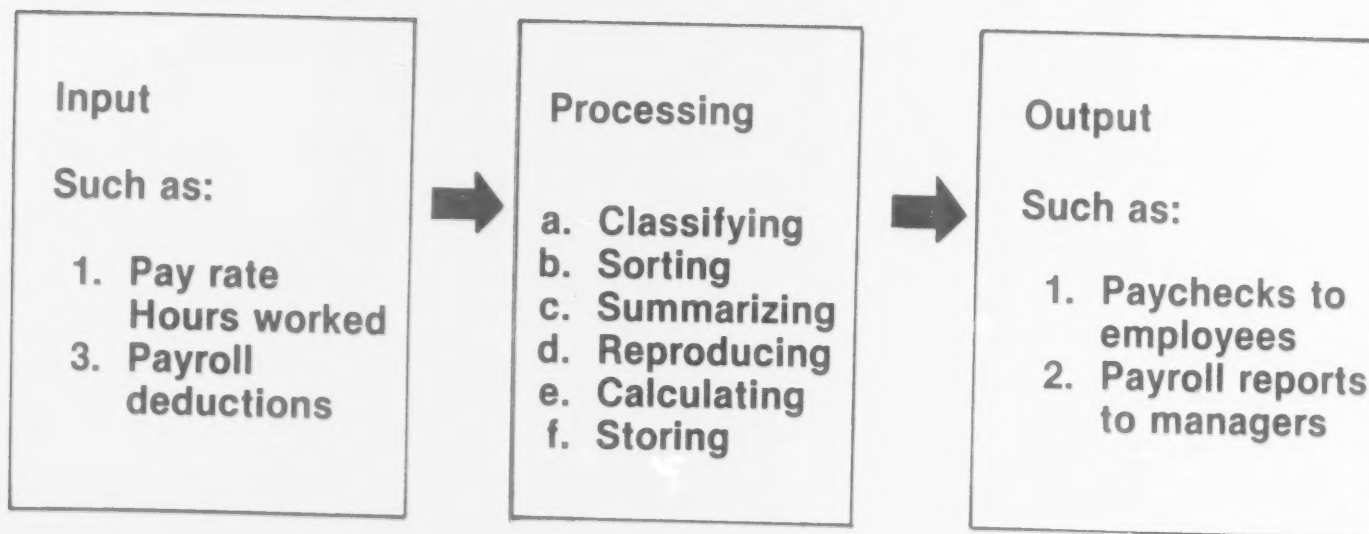
Answer or Desired Result

$$\begin{array}{r} 2 \\ + 2 \\ \hline 4 \end{array}$$

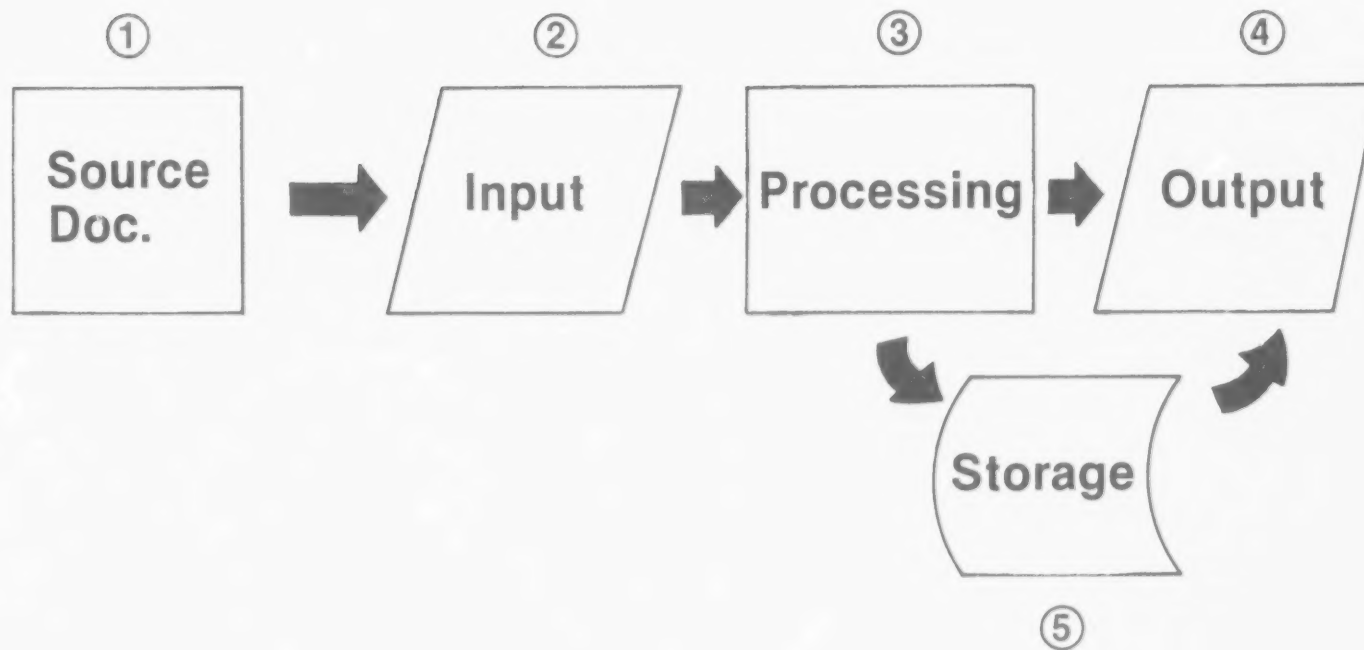
### DATA PROCESSING CYCLE



## Data Processing Functions

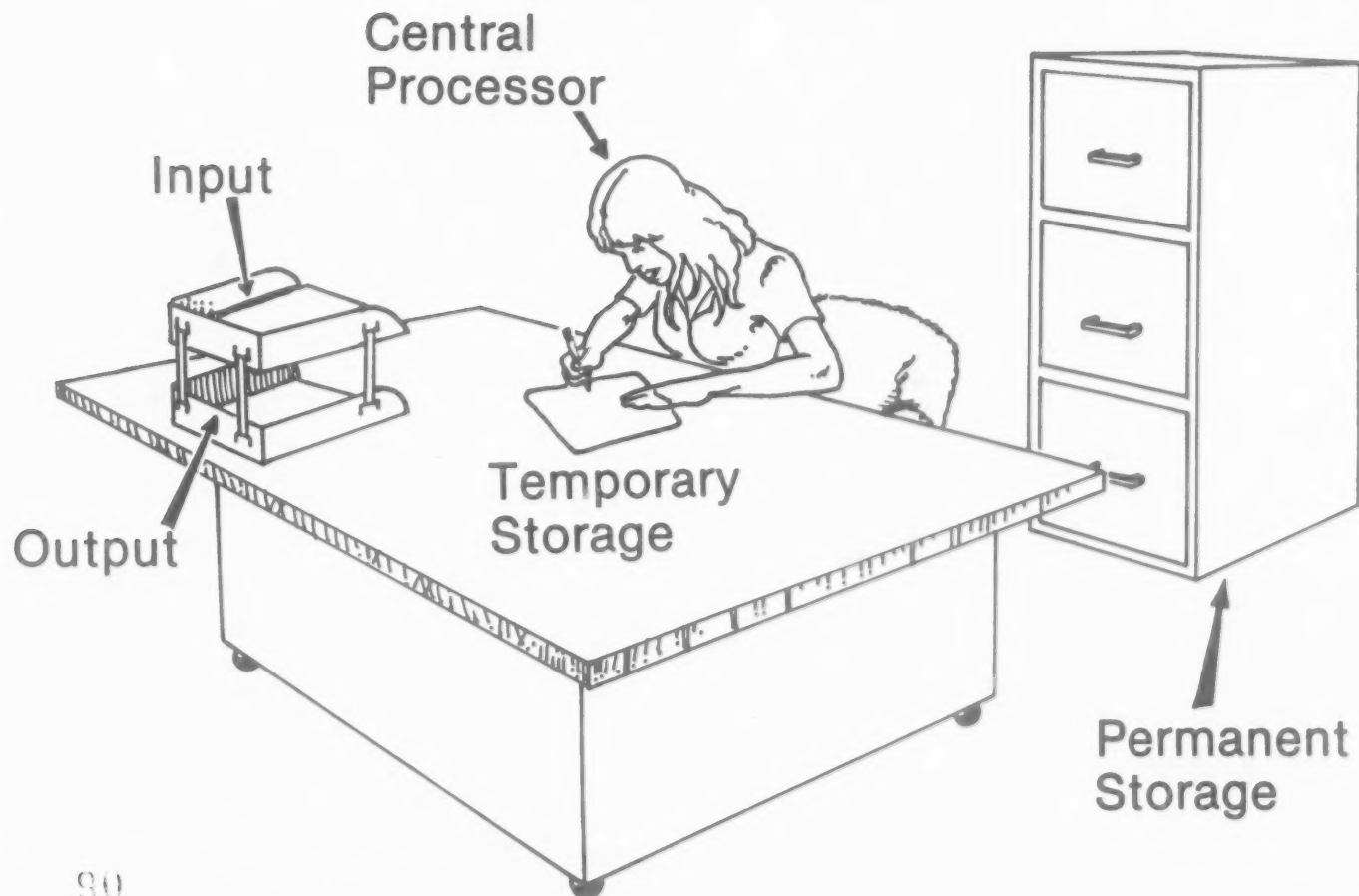


## Components of Information Systems

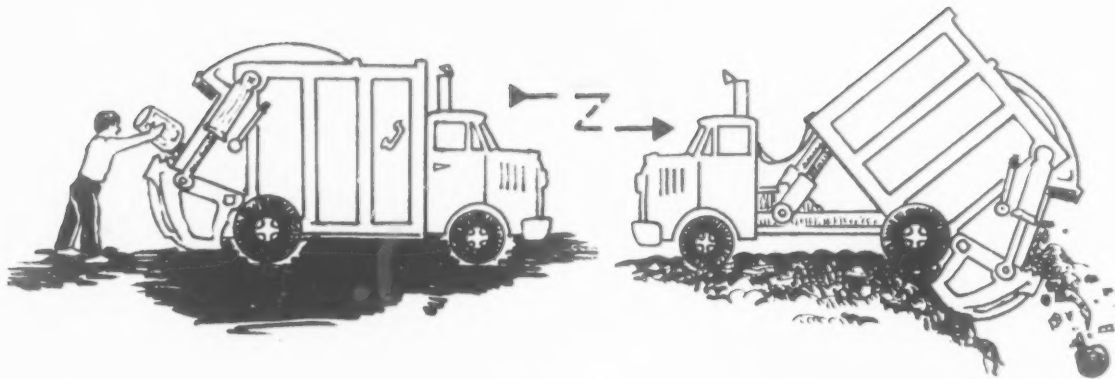




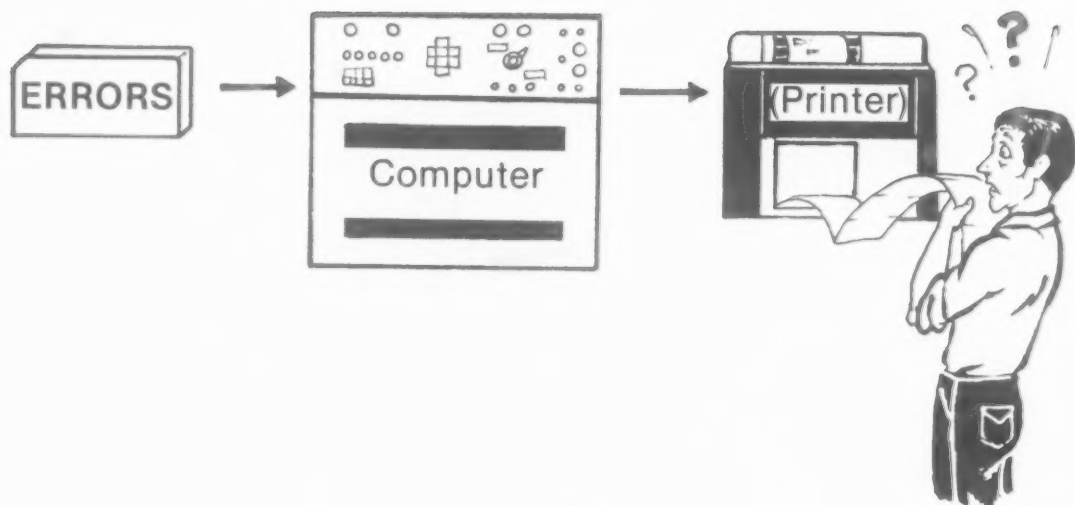
## Components of a Manual Data Processing System



GIGO!



Garbage In = Garbage Out



## SOME SOURCES OF INFORMATION



Time cards



Sales slips

Grade reports

|                 |           |
|-----------------|-----------|
| Bethel Beth 458 |           |
| MALACHI HIGH    |           |
| GRADE REPORT    |           |
| Math            | D         |
| English         | B         |
| French          | C         |
| History         | C         |
| Shop            | B         |
| Typing          | B         |
| TEACHER         | THS. Date |

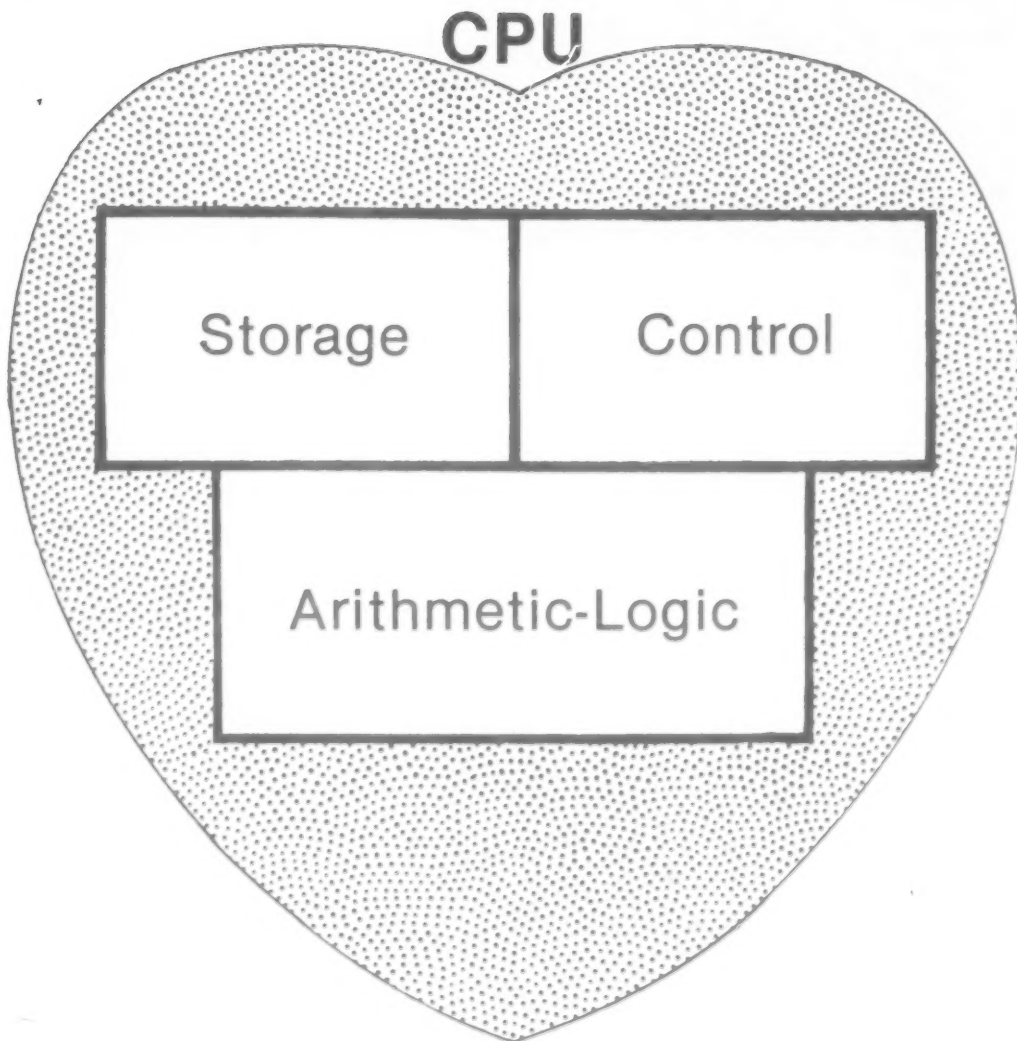
Merchandise tags



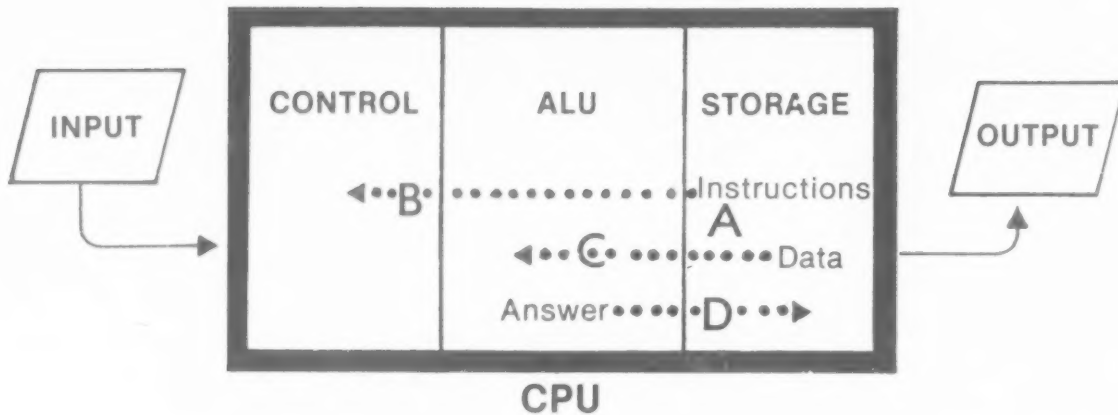
| BANK DEPOSIT FORM   |              |
|---------------------|--------------|
| ACCOUNT NUMBER 2351 | CASH 200.00  |
| NAME Sam Bradley    | CHECKS       |
| FOR BANK USE ONLY   | TOTAL 200.00 |

Bank deposit slips

## THE HEART OF THE COMPUTER SYSTEM

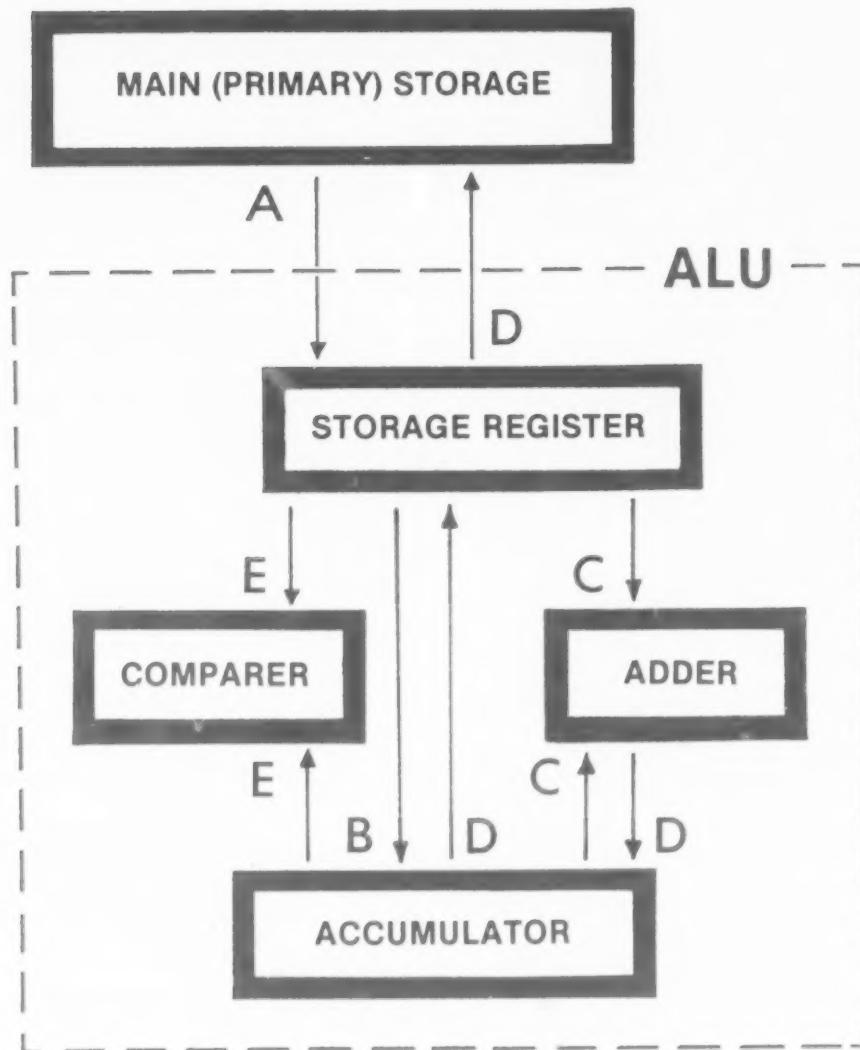


### PROCESSING SEQUENCE



- A. Instructions are stored in the main storage
- B. One instruction is moved to the control unit
- C. Data needed to carry out the instruction are moved to the ALU where the instruction is carried out
- D. The answer to the computation is placed in main storage

### ALU-DATA MOVEMENT



ALU - movement of data through  
Arithmetic-Logic unit

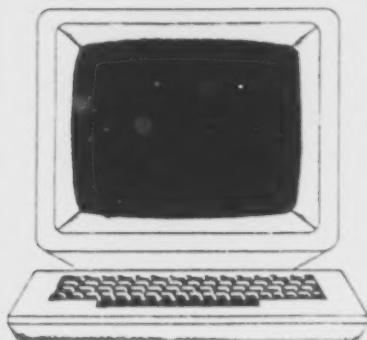
## FORMS OF OUTPUT

### Printed reports



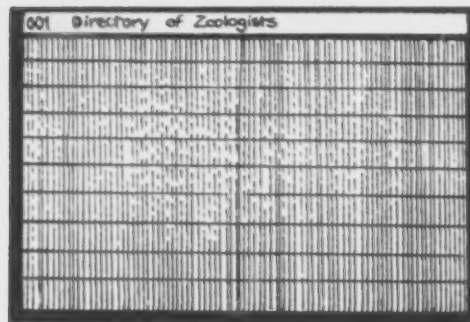
### Checks

|                                                                            |                          |                           |                    |            |
|----------------------------------------------------------------------------|--------------------------|---------------------------|--------------------|------------|
| No. _____                                                                  |                          | NATIONAL BANK             |                    | 1-98<br>21 |
|                                                                            |                          | New York, N.Y. _____ 19__ |                    |            |
| PAY TO THE<br>ORDER OF _____                                               |                          | \$ _____                  |                    |            |
|                                                                            |                          | DOLLARS                   |                    |            |
| ⑆0210⑉098712200842670⑆ <span style="float: right;">⑆042/0000005670⑆</span> |                          |                           |                    |            |
| CHECK<br>ROUTING<br>SYMBOL                                                 | ABA<br>TRANSIT<br>NUMBER | ACCOUNT<br>NUMBER         | PROCESS<br>CONTROL | AMOUNT     |



### Visual

### Microfiche

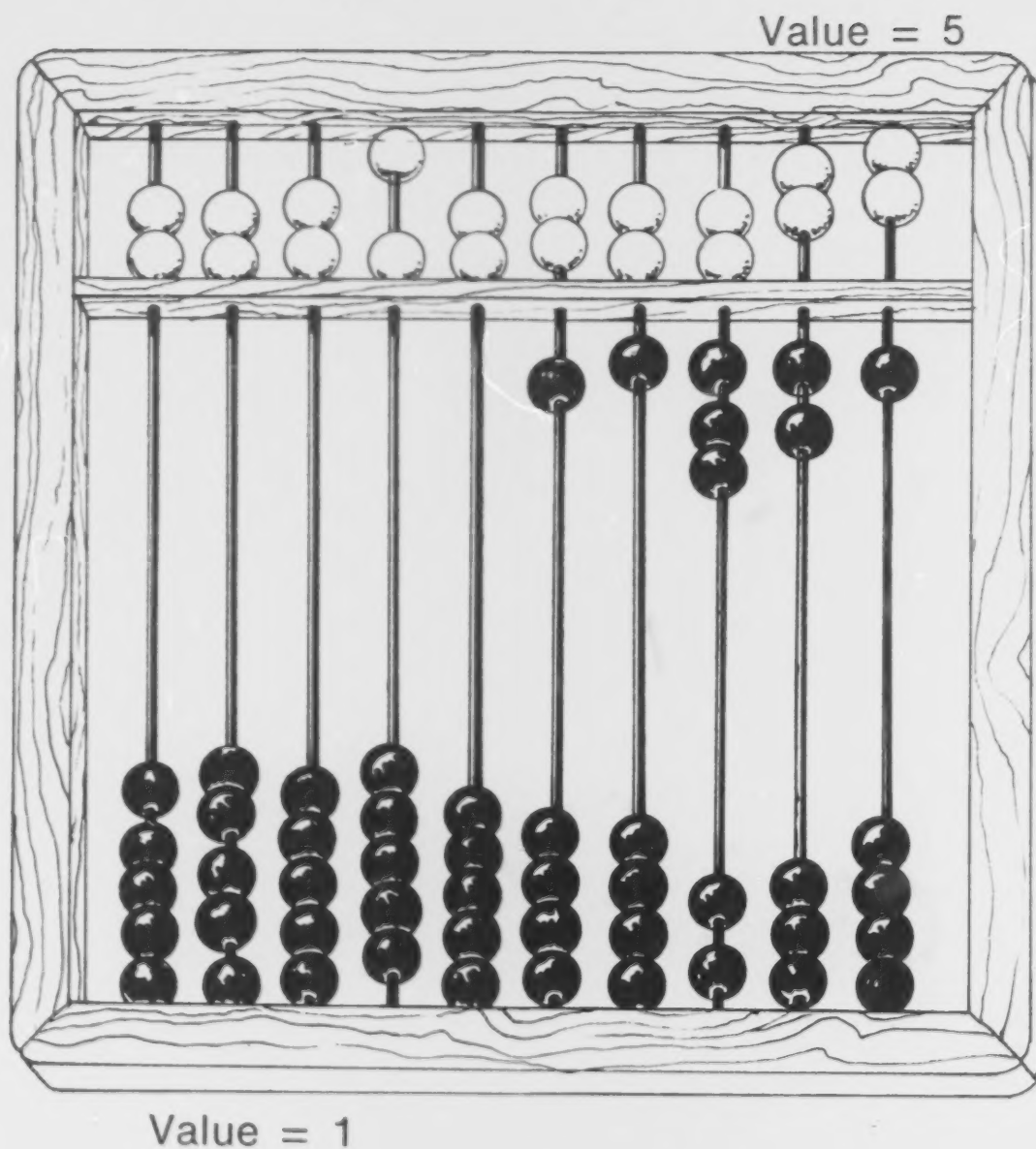


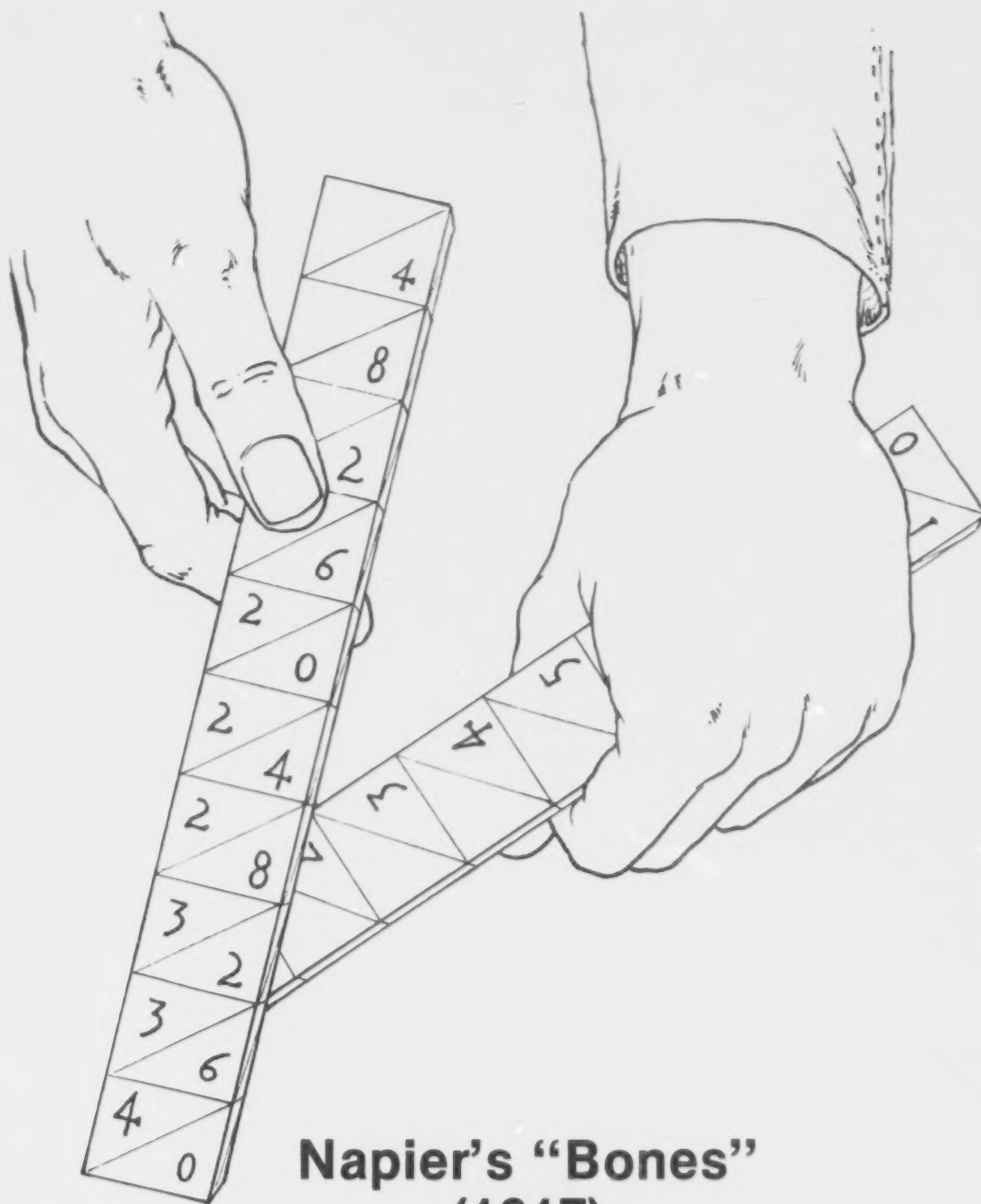
### Microfilm





ABACUS  
(3000 BC)





**Napier's "Bones"  
(1617)**

## SUPPLEMENTARY ACTIVITY

### WHO HAS GAME

**NOTE:** This game may be used with any unit.

**Materials needed:** 30 index cards  
black pen  
word list

#### INSTRUCTIONS:

1. On one side of the index cards, draw a line that splits the cards in half. On one end, write "I HAVE", and on the other end, write "WHO HAS."
2. Take the word list and write "I HAVE" terms on the "I HAVE" end of the index card, and write the "WHO HAS" terms on the other end of the index card.
3. Shuffle all "I HAVE"/"WHO HAS" cards.
4. Deal one card per student until the deck is exhausted.
5. Choose a student to begin the activity by reading the bottom portion of the "WHO HAS" (question).
6. The student in the class having the answer to the "WHO HAS" question at the top of his/her card responds by reading his/her card, "I HAVE" (statement). He/she then reads the bottom portion of the card "WHO HAS" (question).
7. Play continues in this same manner until all terms and definitions have been used. The game ends by returning to the student who began the game.

"I HAVE" categories

- |                 |                       |                       |
|-----------------|-----------------------|-----------------------|
| 1. ASCII        | 11. Auxiliary         | 21. Assembly language |
| 2. EBCDIC       | 12. Direct access     | 22. BASIC             |
| 3. Hardware     | 13. Sequential access | 23. FORTRAN           |
| 4. Peripheral   | 14. Compiler          | 24. COBOL             |
| 5. Software     | 15. Field             | 25. PASCAL            |
| 6. Program      | 16. Record            | 26. RPG               |
| 7. Programmer   | 17. File              | 27. PL/1              |
| 8. User         | 18. Master file       | 28. ADA               |
| 9. Batch        | 19. Detail file       | 29. C                 |
| 10. Transaction | 20. Backup copies     | 30. Machine language  |

"WHO HAS" categories

1. Coding scheme used on mainframes
2. Physical parts of a computer system
3. Input and output devices
4. Computer programs
5. Series of instructions which direct the computer
6. Person who writes programs
7. Person using the computer
8. Processing data as a group
9. Processing data individually
10. Secondary storage
11. Access of data in any order
12. Access of data one record after the other
13. Converts computer programs to machine language
14. One piece of information, such as a date or name
15. A group of fields
16. Group of related records
17. File which contains relatively permanent records
18. File which contains day-to-day transactions
19. Copies made in case the original is damaged
20. Language developed to make machine language easier
21. Primary language for micros
22. Oldest high-level language; used by scientists and mathematicians
23. Language used for business applications developed by the help of Dr. Grace Hopper
24. Used to teach beginners and named after a French inventor
25. Used to write reports
26. Developed by IBM; can be used in science and business applications
27. Used by the U.S. Department of Defense
28. Developed by AT&T; used with special effects in films
29. Difficult language to learn
30. Coding scheme used on microcomputers

## COMPETENCY TEST

### PART I: COMPUTER-RELATED TERMINOLOGY

**INSTRUCTIONS:** Fill in the correct term next to the appropriate definition.

- |       |     |                                                                                                                  |
|-------|-----|------------------------------------------------------------------------------------------------------------------|
| _____ | 1.  | Series of instructions telling the computer what to do.                                                          |
| _____ | 2.  | Physical components of the computer system.                                                                      |
| _____ | 3.  | One piece of information, such as a name or date.                                                                |
| _____ | 4.  | Copies made in case the original is damaged.                                                                     |
| _____ | 5.  | Used in measuring computer memory; equals 1024 bytes.                                                            |
| _____ | 6.  | Access when data can be retrieved in any order.                                                                  |
| _____ | 7.  | Coding scheme used on microcomputers.                                                                            |
| _____ | 8.  | Device used to connect computers through phone lines.                                                            |
| _____ | 9.  | Equals one character.                                                                                            |
| _____ | 10. | Person who writes programs.                                                                                      |
| _____ | 11. | A program that interprets computer statements written in another language and converts them to machine language. |
| _____ | 12. | A group of fields.                                                                                               |
| _____ | 13. | Access when data are retrieved one record after the other.                                                       |
| _____ | 14. | All instructions and information necessary to make the computer function.                                        |
| _____ | 15. | A term used when referring to input and output devices.                                                          |
| _____ | 16. | A binary digit, zero or one.                                                                                     |
| _____ | 17. | A term used for secondary storage.                                                                               |
| _____ | 18. | The accumulating and processing of data as a group.                                                              |
| _____ | 19. | A file that contains relatively permanent records.                                                               |

- |       |     |                                                            |
|-------|-----|------------------------------------------------------------|
| _____ | 20. | A group of related records.                                |
| _____ | 21. | Computers which work by numbers and a set of instructions. |
| _____ | 22. | Processing data one record at a time.                      |
| _____ | 23. | Computers that measure physical conditions.                |
| _____ | 24. | Transmission of data from one computer to another.         |
| _____ | 25. | One letter, number, or symbol.                             |

## PART II: FUNCTIONS OF DATA PROCESSING

**INSTRUCTIONS:** Define each function of data processing.

1. Data collecting
2. Recording
3. Sorting
4. Classifying
5. Calculating
6. Storing and retrieving
7. Summarizing
8. Communicating



### PART III: PARTS OF THE INFORMATION PROCESSING CYCLE

#### Project 1

**INSTRUCTIONS:** List and describe each step of the information processing cycle.

1.

2.

3.

4.

### PART III (Continued)

#### Project 2

**INSTRUCTIONS:** List two parts of the processor and two parts of the CPU.

Processor

1.

2.

CPU

1.

2.

### PART III (Continued)

#### Project 3

INSTRUCTIONS: Draw the parts of the computer processor.

## PART IV: COMPARISON OF COMPUTER SYSTEMS

**INSTRUCTIONS:** Describe each computer system using at least three characteristics discussed in class.

1. Supercomputers
2. Mainframes
3. Minicomputers
4. Microcomputers
5. Laptop computers

## PART V: EVENTS IN HISTORY

### Project 1

**INSTRUCTIONS:** Select the appropriate answer for each of the questions concerning the history of data processing. Place your answer in the space provided.

- \_\_\_\_\_ 1. This device was not a computer, but it used data processing techniques in the 1800s.
- |                   |                    |
|-------------------|--------------------|
| A. Napier's bones | C. Jacquard's loom |
| B. Abacus         | D. Leibnitz wheel  |
- \_\_\_\_\_ 2. This machine was never built, but the concepts in the design were similar to the concepts of today's computers.
- |                      |                   |
|----------------------|-------------------|
| A. Analytical engine | C. Leibnitz wheel |
| B. ENIAC             | D. Abacus         |
- \_\_\_\_\_ 3. The Orientals used this as a counting device.
- |                   |                         |
|-------------------|-------------------------|
| A. Napier's bones | C. Leibnitz wheel       |
| B. Abacus         | D. Machine arithmetique |
- \_\_\_\_\_ 4. The counting of the U.S. Census was done with the help of this device.
- |                             |           |
|-----------------------------|-----------|
| A. Analytical engine        | C. Mark I |
| B. Hollerith's punched card | D. ENIAC  |
- \_\_\_\_\_ 5. This calculator operated by turning dials.
- |                         |                    |
|-------------------------|--------------------|
| A. Machine arithmetique | C. Abacus          |
| B. Napier's bones       | D. Jacquard's loom |
- \_\_\_\_\_ 6. Considered to be the "Father of Computers."
- |                     |                    |
|---------------------|--------------------|
| A. Herman Hollerith | C. Joseph Jacquard |
| B. Blaise Pascal    | D. Charles Babbage |
- \_\_\_\_\_ 7. Considered to be the first programmer.
- |                    |                  |
|--------------------|------------------|
| A. Charles Babbage | C. Blaise Pascal |
| B. Ada Lovelace    | D. John Napier   |

**PART V (Continued)**

**Project 1 (Continued)**

- \_\_\_\_\_ 8. The first computer to be used commercially.
- |             |           |
|-------------|-----------|
| A. ENIAC    | C. Mark I |
| B. UNIVAC I | D. EDSAC  |
- \_\_\_\_\_ 9. Considered to be the first electronic computer.
- |             |          |
|-------------|----------|
| A. Mark I   | C. EDSAC |
| B. UNIVAC I | D. ENIAC |
- \_\_\_\_\_ 10. First machine to use the stored-program concept.
- |             |          |
|-------------|----------|
| A. Mark I   | C. EDSAC |
| B. UNIVAC I | D. ENIAC |
11. Write a paragraph about the five computer generations, distinguishing each generation from the other.

## PART V (Continued)

### Project 2

**INSTRUCTIONS:** Place the appropriate letter in the blank provided.

- |                     |                 |                    |
|---------------------|-----------------|--------------------|
| A. Charles Tandy    | F. Jack Kilby   | K. Jack Tramiel    |
| B. Nolan Bushnell   | G. John Kemeny  | L. IBM             |
| C. Stephen Wozniak  | H. Grace Hopper | M. Adam Osborn     |
| D. Herman Hollerith | I. Steve Jobs   | N. Mitch Kapor     |
| E. H. Ross Perot    | J. Thomas Kurtz | O. Daniel Bricklin |
|                     |                 | P. VisiCalc        |

- \_\_\_\_\_ 1. Navy commander who contributed to the writing of the COBOL programming language.
- \_\_\_\_\_ 2. The Hollerith company was purchased and its name was changed to \_\_\_\_\_.
- \_\_\_\_\_ 3. Invented a punched card process used in the 1890 census.
- \_\_\_\_\_ 4. Turned a hobby and supplies business into an electronic conglomerate.
- \_\_\_\_\_ 5. Founder of Electronic Data Systems.
- \_\_\_\_\_ 6. Developed the most widely used spreadsheet program, Lotus 1-2-3.
- \_\_\_\_\_ 7. Responsible for developing the first BASIC program in 1964.
- \_\_\_\_\_ 8. Introduced the Commodore series of computers.
- \_\_\_\_\_ 9. Introduced a computer that will fit under an airplane seat.
- \_\_\_\_\_ 10. One of the men who produced the original Apple computer.
- \_\_\_\_\_ 11. The first electronic spreadsheet program.
- \_\_\_\_\_ 12. Developed the NeXt Company for applications in a college environment.
- \_\_\_\_\_ 13. Invented the microchip.
- \_\_\_\_\_ 14. Wrote the first electronic spreadsheet.
- \_\_\_\_\_ 15. Known as the entrepreneur who began the Atari video game computer.

## PART VI: COMPUTER LANGUAGES

### Project 1

**INSTRUCTIONS:** List six categories of languages.

1.

2.

3.

4.

5.

6.



## PART VI (Continued)

### Project 2

**INSTRUCTIONS:** Explain why high-level languages were developed and how they benefit the programmer.

## PART VI (Continued)

### Project 3

**INSTRUCTIONS:** Match the language with the appropriate descriptions.

| LANGUAGE         | DEFINITIONS                                                                          |
|------------------|--------------------------------------------------------------------------------------|
| _____ 1. BASIC   | A. Language experiencing a rapid rate of growth; named after a French inventor.      |
| _____ 2. FORTRAN | B. IBM's answer to languages with shortcomings.                                      |
| _____ 3. RPG     | C. Language used primarily on micros and that is good to teach beginners.            |
| _____ 4. COBOL   | D. Developed by U.S. Department of Defense because of their need of standardization. |
| _____ 5. PASCAL  | E. Generates reports.                                                                |
| _____ 6. ADA     | F. Used mainly for business; Grace Hopper helped develop this language.              |
| _____ 7. C       | G. Used primarily for solving scientific, engineering, and math problems.            |
| _____ 8. FORTH   | H. Used for industrial process control applications.                                 |
| _____ 9. PL/1    | I. Low-level language developed by AT&T.                                             |

## ANSWERS TO ACTIVITIES

### Activity 1

|               |             |       |
|---------------|-------------|-------|
| MENU          | F           | P     |
|               | EBCDIC      | R     |
| A C A         | E           | O     |
| OUTPUT        | L           | G     |
| S X U C       | D           | R     |
| O I H         | D           | A     |
| FILE          | D KILOBYTE  | M     |
| T I B         | I G A       | C M   |
| W ASCII       | R I C       | O E   |
| A R D         | E T K       | R R   |
| R Y           | C A U       | D     |
| E SEQUENTIAL  | P           | C     |
| T             | H           | O     |
| MODEM         |             | MODEM |
| R             | R           | P     |
| ANALOG        | D           | I     |
| G             | W           | L     |
| USER          | TRANSACTION | E     |
|               | R           | R     |
| DATASTRUCTURE |             |       |

### Activity 2

1. **Summarizing**--condensing data and information.
2. **Calculating**--performing mathematical operations.
3. **Data collecting**--collecting raw data that need to be processed.
4. **Sorting**--putting data in ascending /descending alphabetical or numeric order.
5. **Communicating**--transferring data from one place to another.
6. **Recording**--putting data in some form that is recognizable by human or machine.
7. **Storing and retrieving**--storing data on some type of media for future use.
8. **Classifying**--arranging data according to some logical relationship.

### Activity 3

#### Project 1

The diagram of the information processing cycle should include the following, in this order:

**input**--putting raw data into the computer to be processed;  
**processing**--manipulating the data to get a desired result;  
**output**--communicating the answer to the user; and  
**storage**--where the data is kept outside of the computer memory for future use.

## Project 2

Processing cycle designs should include the following steps:

1. Gathering information  
Arranging information  
Putting the completed research on paper
2. Gathering ingredients  
Mixing and/or cooking ingredients  
Serving the finished product, such as a cake or hamburger
3. Gathering miles per gallon and amount of gas used  
Dividing miles per gallon by amount of gas used  
Displaying the final mileage total
4. Gathering the rate of pay and hours worked  
Multiplying rate of pay times hours worked and subtracting deductions  
Displaying the final salary total
5. Gathering and recording grades  
Adding grades and dividing by the total number of grades  
Displaying the final answer
6. Interviewing applicants  
Comparing applicants and choosing the one most qualified  
Hiring the applicant

Project 3 The diagram of the processor should resemble suggested Transparency 9.

## Project 4

- |    |   |    |   |
|----|---|----|---|
| 1. | A | 4. | D |
| 2. | D | 5. | C |
| 3. | C | 6. | B |

## Activity 4

Project 1 Answers will vary according to the instructor's lecture and textbook used.  
Evaluated to the satisfaction of the instructor.

## Project 2

- |    |   |    |   |    |   |     |   |
|----|---|----|---|----|---|-----|---|
| 1. | B | 4. | E | 7. | B | 10. | C |
| 2. | D | 5. | D | 8. | E | 11. | D |
| 3. | C | 6. | C | 9. | B | 12. | A |

Project 3 This report will vary according to the six employees interviewed by the student. Evaluated to the satisfaction of the instructor.

**Project 4** Computer system posters will be evaluated to the satisfaction of the instructor.

**Activity 5**

**Project 1**

- |    |   |    |   |     |   |
|----|---|----|---|-----|---|
| 1. | F | 5. | J | 9.  | B |
| 2. | G | 6. | A | 10. | I |
| 3. | D | 7. | L | 11. | C |
| 4. | K | 8. | E | 12. | H |

**NOTE:** Dates may vary depending on textbook used.

**Project 2** A time line showing and describing each development in computer history will be evaluated to the satisfaction of the instructor.

**Project 3** Answers will vary depending on the person chosen. Report evaluated to the satisfaction of the instructor.

**Project 4** Evaluated to the satisfaction of the instructor.

**Activity 6**

**Project 1**

- |     |                                    |           |    |             |
|-----|------------------------------------|-----------|----|-------------|
| 1.  | 1.                                 | Level     | 4. | Orientation |
|     | 2.                                 | Purpose   | 5. | Translation |
|     | 3.                                 | Structure | 6. | Interaction |
| 2.  | COBOL, PASCAL in the future, PL/1. |           |    |             |
| 3.  | RPG                                |           |    |             |
| 4.  | C                                  |           |    |             |
| 5.  | APL                                |           |    |             |
| 6.  | COBOL                              |           |    |             |
| 7.  | FORTRAN                            |           |    |             |
| 8.  | BASIC, PASCAL, FORTH               |           |    |             |
| 9.  | FORTH                              |           |    |             |
| 10. | ADA                                |           |    |             |
| 11. | Assembly                           |           |    |             |

Project 2

Five of the following:

1. COBOL
2. FORTRAN
3. BASIC or PASCAL
4. ADA
5. RPG

Project 3

High-level languages enable the programmer to focus on the problem rather than on details of computer operations. In addition, high-level languages may be used with different makes of computers. Points that might be included in the answer:

They are easier to learn than assembly languages.  
They require less time to write.  
They provide better documentation.  
They are easier to maintain.

## COMPETENCY TEST ANSWERS

### Part I

- |                  |                         |
|------------------|-------------------------|
| 1. Program       | 14. Software            |
| 2. Hardware      | 15. Peripheral          |
| 3. Field         | 16. Bit                 |
| 4. Backups       | 17. Auxiliary           |
| 5. K             | 18. Batch               |
| 6. Random/direct | 19. Master              |
| 7. ASCII         | 20. Files               |
| 8. Modem         | 21. Digital             |
| 9. Byte          | 22. Transaction         |
| 10. Programmer   | 23. Analog              |
| 11. Compiler     | 24. Data communications |
| 12. Record       | 25. Character           |
| 13. Sequential   |                         |

### Part II

1. Data collecting is collecting raw data that need to be processed.
2. Recording is putting data in some form that is recognizable by person or machine.
3. Sorting is putting data in ascending/descending alphabetical or numerical order.
4. Classifying is arranging data according to some logical relationship.
5. Calculating is performing mathematical operations.
6. Storing and retrieving is putting data on some type of media for future use.
7. Summarizing is condensing data and information.
8. Communicating is transferring data from one place to another.

### Part III

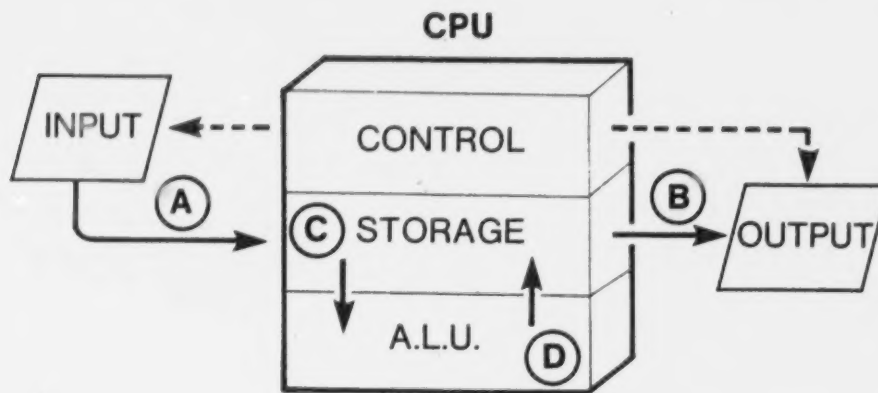
#### Project 1

1. **Input**--entering raw data into the computer to be processed
2. **Process**--manipulating data to achieve a desired result
3. **Output**--communicating the result to the user
4. **Storage**--storing programs and data outside of the main memory for future use

#### Project 2

- |           |    |                       |
|-----------|----|-----------------------|
| Processor | 1. | Arithmetic/logic unit |
|           | 2. | Control unit          |
| CPU       | 1. | Read only memory      |
|           | 2. | Random access memory  |

**Project 3**



**Part IV** Evaluated to the satisfaction of the instructor.

**Part V**

**Project 1**

- |      |      |      |       |
|------|------|------|-------|
| 1. C | 4. B | 7. B | 10. A |
| 2. A | 5. A | 8. B |       |
| 3. B | 6. D | 9. D |       |
11. Evaluated to the satisfaction of the instructor.

**Project 2**

- |      |       |       |
|------|-------|-------|
| 1. H | 6. N  | 11. D |
| 2. L | 7. G  | 12. I |
| 3. D | 8. K  | 13. F |
| 4. A | 9. M  | 14. O |
| 5. E | 10. C | 15. B |

**Part VI**

**Project 1**

- |              |                |
|--------------|----------------|
| 1. Level     | 4. Orientation |
| 2. Purpose   | 5. Translation |
| 3. Structure | 6. Interaction |

120



**Project 2**

High-level languages enable the programmer to focus on the problem rather than on details of computer operations. In addition, high-level languages may be used with different makes of computers. Points that might be included in the answer:

They are easier to learn than assembly languages.  
They require less time to write.  
They provide better documentation.  
They are easier to maintain.

**Project 3**

- |    |   |    |   |    |   |
|----|---|----|---|----|---|
| 1. | C | 4. | F | 7. | I |
| 2. | G | 5. | A | 8. | H |
| 3. | E | 6. | D | 9. | B |

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(1)(A)(B)

The student will be provided opportunities to demonstrate comprehension of computer-related terms; and identify the major components of computer systems.

| OBJECTIVE: | Students will demonstrate the ability to understand computer processing concepts.                                      | MASTERY | NON-MASTERY |
|------------|------------------------------------------------------------------------------------------------------------------------|---------|-------------|
| 1.         | Identify basic computer-related terminology.                                                                           | _____   | _____       |
| 2.         | Outline the functions of data processing.                                                                              | _____   | _____       |
| 3.         | Illustrate the parts of the information processing cycle and explain how each part is interdependent within the cycle. | _____   | _____       |
| 4.         | Illustrate parts of the computer processor.                                                                            | _____   | _____       |
| 5.         | Compare the similarities and differences among various computer systems.                                               | _____   | _____       |
| 6.         | Outline the events in history pertinent to breakthroughs in data processing.                                           | _____   | _____       |
| 7.         | Define and categorize characteristics of computer languages.                                                           | _____   | _____       |
|            |                                                                                                                        | _____   | _____       |
|            |                                                                                                                        | _____   | _____       |

STUDENT NAME \_\_\_\_\_

122

## SECTION I: INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING I

### UNIT I: INTRODUCTION TO THE COURSE

#### UNIT

#### OBJECTIVE:

Students will develop an understanding of the purpose, contents, and organization of the course Business Computer Programming I.

#### STUDENT

**COMPETENCIES:** After completing this unit, the student will be able to:

1. Relate the purpose of Business Computer Programming I to office education, business, and industry.
2. Recognize the role of the youth organization relative to Business Computer Programming I.
3. Discuss course objectives, equipment, units, competencies, and evaluation measures.
4. Explain classroom organization, management, and safety.

#### FOCUS:

Students will be provided the opportunity to develop and use both academic and vocational skills that will be useful whether they are taking advanced courses or making the transition from high school to the world of work.

## TERMINOLOGY

|                                          |                                                                                                                                       |
|------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Business computer programming</b>     | An occupationally specific course under the umbrella of office education for students to learn and apply computer programming skills. |
| <b>Business Professionals of America</b> | A youth organization designed to provide opportunities for growth and development of the business, office, and leadership skills.     |
| <b>Classroom management</b>              | Policies and procedures for general classroom operations which enhance the learning environment.                                      |
| <b>Classroom organization</b>            | Arrangement of classroom machines, equipment, furniture, or seating which provides for optimum instruction.                           |
| <b>Cluster course</b>                    | A foundation course which includes concepts and skills that are needed in office occupations.                                         |
| <b>Competency</b>                        | The ability to do something.                                                                                                          |
| <b>Competency checklist</b>              | A listing of competencies indicating mastery or nonmastery of each.                                                                   |
| <b>Evaluation measures</b>               | Methods for evaluating student performance.                                                                                           |
| <b>Occupational specific course</b>      | A course which develops skills needed to perform a specific occupation.                                                               |
| <b>Office education</b>                  | One of the program areas which provides for vocational education in office-related occupations.                                       |
| <b>Safety precautions</b>                | Following and applying safety awareness in an office environment.                                                                     |
| <b>Units of instruction</b>              | Identification of specific topics to be taught during a course.                                                                       |

## UNIT OVERVIEW

### I. Purpose of course

A. Business Computer Programming I is one of the occupational specific courses coming under the umbrella of office education. Office education is one of several vocational-technical program areas which combines education and job training.

1. The main goal of this office education course is to deliver instruction that will provide the student with:
  - a. Marketable skills.
  - b. Knowledge.
  - c. Attitudes that will be useful upon graduation.
2. In addition to this primary goal, office education benefits students in other ways by:
  - a. Assisting in making the transition from high school to the world of work.
  - b. Providing experiences for developing leadership skills.
  - c. Assisting with making career choices.
  - d. Developing skills, knowledge, and attitudes needed in assuming adult responsibilities.
3. Students enrolled in Business Computer Programming I have an opportunity to participate in youth organization activities at the local, state, and national levels through Business Professionals of America. Business Professionals of America encourages students to participate in state and national conferences organized to enhance leadership skills. It is the **only** approved student organization for students enrolled in Business Computer Programming I.
  - a. History of Business Professionals of America
  - b. Competitive events
    - (1) Events consist of various contests which provide opportunities for competition with students from other schools.
    - (2) Students learn sportsmanship by:

- (a) Being a gracious winner.
    - (b) Being a gracious loser.
  - c. Leadership opportunities
    - (1) Opportunities are available to run for state or national office.
    - (2) To be elected as an officer by one's peers is an honor.
    - (3) With honor comes responsibility, this means:
      - (a) Putting the organization first.
      - (b) Speaking in front of groups.
      - (c) Planning conferences.
- B. Objectives of Business Computer Programming I
  - 1. The content of this course has been developed from the essential elements required by the State Board of Education.
  - 2. The objectives of these essential elements are to provide learning experiences that will enable students to acquire competency in performing basic tasks relative to office occupations.
- C. Relationship of Business Computer Programming I to business--This course has been developed to provide fundamental programming concepts and skills needed by general office workers.
  - 1. This course is considered a foundation course; that is, the content would be useful in any office position. It is composed of instructional activities that will be helpful in getting an office job or in providing the foundation needed before taking more advanced computer-related courses.
  - 2. Job market projections indicate that office workers will continue to be in demand.
    - a. Regardless of the type of business, people will be needed to handle the computer operations of the company.
    - b. Chances for advancement are good for the worker who has computer related skills.
    - c. Some companies offer incentives such as paying part or all of tuition, fees, and books for employees who further their education. Business Computer Programming I courses are considered a steppingstone into the business world.

## II. Course overview

- A. **Classroom equipment**--the office equipment needed for Business Computer Programming I has been determined by the essential elements and by the course competencies. The equipment reflects the current skills needed by the beginning office worker.
- B. **Units**--Business Computer Programming I includes twelve units, falling under the following sections:
  - 1. **Introduction to Business Computer Programming I**--review of purposes, information and activities, classroom organization, evaluation techniques, social implications, and computer-related operations.
  - 2. **Computer hardware utilization**--computer hardware, computer software, and computer operation and maintenance.
  - 3. **Program development**--problem-solving concepts and applications and programming concepts and applications.
  - 4. **Data storage**--media and types of storage, data storage applications, and data representation and numbering systems.
  - 5. **Closure of course**--review of course, with emphasis on competencies gained from the course.
- C. **Competencies**--the ability to do something. The ability to do something includes attitudes, values, skills, and/or knowledge. Most of the competencies included in this course are job-related. This means that these competencies are needed to perform an office job.
  - 1. A competency checklist is included at the end of each unit.
  - 2. All competencies need to be reviewed before beginning unit lessons.
- D. **Evaluation**--a variety of methods are used to evaluate performance and knowledge. Some of the methods might include:
  - 1. In-class activities.
  - 2. Outside assignments.
  - 3. Skill production.
  - 4. Tests.
- E. **Classroom organization and management**



1. The organization of the classroom has been established to provide optimum experiences for learning. Class size and number of machines are determining factors in classroom organization.
2. For management purposes, it is necessary to establish policies and/or procedures to ensure quality of performance. For example:
  - a. Care of equipment
  - b. Use of supplies

F. Safety

1. Office workers often feel secure when it comes to safety. However, the office worker should take responsibility for learning more about the office setting and the office job itself. The beginning worker does not yet have the experiences to realize the potential danger involved.
2. Companies give much thought and planning to developing a safe, secure environment for their employees. Many elements are taken into consideration to provide facilities and programs that will contribute to the well-being of the people who work in offices.
3. One of the areas of concentration is office design, which involves:
  - a. Attractive and efficient surroundings.
  - b. Good lighting for sufficient illumination for the work area.
  - c. Special materials on floors and walls to absorb loud or irritating noises caused by office equipment.
4. Ninety percent of office accidents result from unsafe actions or conditions:
  - a. Slipping, tripping, or falling
  - b. Improper use of equipment
  - c. Obstructions
  - d. Fire
  - e. Electricity
5. The federal government has developed special guidelines to provide for safe and healthful work environments. However, employees must do their part. Some of the things employees can do to contribute to office safety are as follows:
  - a. Use good judgment.



- b. Observe and follow safety procedures.
  - c. Use safety precautions.
  - d. Avoid horseplay while on the job.
  - e. Avoid becoming too emotional which could lead to loss of control.
  - f. Do not use alcoholic beverages or drugs.
  - g. Avoid overloading electrical circuits.
  - h. Keep walkways clear.
  - i. Report frayed or damaged electrical cords.
  - j. Close desk, file, or other drawers before leaving the work area.
  - k. Immediately unplug any machine or electrical appliance which smells like smoke.
  - l. Be careful when sitting in rolling chairs or one that leans back too far.
  - m. Avoid unsafe dress and grooming practices.
  - n. Have proper ventilation.
  - o. Be aware that sharp-pointed objects are hazardous.
6. **In case of an accident**--the office worker should know the company policies regarding injuries or accidents. The following steps are usually part of the procedures to follow:
- a. Turn off power.
  - b. Administer first aid.
  - c. Report all accidents and injuries.
  - d. Take injured person to hospital emergency room.
7. **Fire safety**--Some general procedures should be followed relative to fire prevention:
- a. Report a fire quickly.
  - b. Know the types of extinguishers.

- c. Know location of fire extinguishers and how to use them.

(WARNING: DO NOT play with firefighting equipment, because it may not work when you need it.)

- d. Know the location of the fire alarm, or notify the office.
- e. Keep a clean work area with flammables stored properly.
- f. Know the correct way to leave a building in case of a fire.

## SECTION I: INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING

### UNIT II: SOCIAL IMPLICATIONS

#### UNIT

#### OBJECTIVE:

Students will describe the issues related to the responsible use of computer systems, appraise the social ramifications of computer usage by society, and identify responsible practices in the use of computers.

#### STUDENT

**COMPETENCIES:** After completing this unit, the student will be able to:

1. Identify terminology related to computer crime, computer users, and new computer devices and techniques.
2. Identify the major issues and technological developments related to computers and society.
3. Describe the ethical and legal issues that have arisen as a result of the computer revolution.
4. Identify privacy issues involved with electronic data storage.
5. Explain the reasons for concern about the reliability of computer systems.

#### FOCUS:

Using a computer is like using a piece of powerful machinery such as an automobile. Most people use computers harmlessly and responsibly, but if the user is irresponsible, careless, or malicious, the effects of decisions and actions can be far-reaching and damaging. Computers have caused so many changes in the way we live and work that society is totally dependent upon them (as it is on the automobile). The first step toward responsible use of computers is understanding the scope and implications of the relationship between society and technology.

## TERMINOLOGY

|                                     |                                                                                                                                                                                                 |
|-------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Activation</b>                   | The final phase of the virus life cycle.                                                                                                                                                        |
| <b>Activation period</b>            | The time delay between initial system infection and virus activation.                                                                                                                           |
| <b>Alameda virus</b>                | Replaces the original boot sector with itself; causes loss of data.                                                                                                                             |
| <b>Artificial Intelligence (AI)</b> | A field of study involving techniques under which computers can formulate responses on the basis of content of accumulated and experience-related information.                                  |
| <b>CD ROM</b>                       | The current compact disks for computers, similar to read-only memory; they cannot be erased or recorded on.                                                                                     |
| <b>Computer crime</b>               | Any illegal act for which knowledge of computer technology is used to commit the offense.                                                                                                       |
| <b>Computer ethics</b>              | A term used to refer to the standard of moral conduct in computer use; a way in which the "spirit" of some laws are applied to computer-related activities.                                     |
| <b>Computer virus</b>               | A segment of machine-language code that replaces software or data with a copy of itself without authorization.                                                                                  |
| <b>Datacrime</b>                    | A virus that breaks down the directory section, making it impossible for a disk's contents to be read.                                                                                          |
| <b>Data spying</b>                  | A form of illegal access to data.                                                                                                                                                               |
| <b>Decision support system</b>      | An information-producing system aimed at a particular decision a manager must make.                                                                                                             |
| <b>Encryption</b>                   | A term describing data that has been translated into a secret code for security reasons.                                                                                                        |
| <b>Expert system</b>                | A program that mimics the decision-making logic of human experts.                                                                                                                               |
| <b>Hacker</b>                       | A term used to describe computer enthusiasts who are challenged by the practice of breaking computer security measures designed to prevent unauthorized access to a particular computer system. |

|                              |                                                                                                                                                                                                        |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Hidden virus</b>          | A virus that leaves no external signs of its presence after infection.                                                                                                                                 |
| <b>Host program</b>          | The program to which the virus attaches itself.                                                                                                                                                        |
| <b>Hypergraphics</b>         | A method used in classrooms or in-service training that involves the participants, the teacher, and the computer through the use of a response pad and an interconnection box.                         |
| <b>Infection</b>             | The invasion of a virus into a host system.                                                                                                                                                            |
| <b>Isolation</b>             | The process a virus uses to identify and distinguish itself from its host program.                                                                                                                     |
| <b>Israeli virus</b>         | Infects any .COM or .EXE program; some versions destroy all data on the hard disk; stored programs will disappear.                                                                                     |
| <b>Lehigh virus</b>          | Infects COMMAND.COM file; causes loss of all data on hard disk.                                                                                                                                        |
| <b>Logic Bomb</b>            | A virus that checks for a certain action, such as if an infected program has been executed a preset number of times.                                                                                   |
| <b>Network worms</b>         | Programs that use network connections to spread from system to system, thus network worms attack systems that are linked via communication lines. It can behave like a virus, or it can plant a virus. |
| <b>nVIR</b>                  | Appears in many different varieties, each with its own individual activation characteristics. Causes loss of data and programs; causes frequent system crashes.                                        |
| <b>Pakistani Brain virus</b> | Replaces original boot sector with itself; system crash can cause loss of data; spreads quickly to all bootable disks.                                                                                 |
| <b>Paperless office</b>      | A term that implies that all paper documents have been replaced by electronic media and devices.                                                                                                       |
| <b>Peace virus</b>           | A type of virus that was created to spread a greeting of world peace and then erases its own instructions, disappearing without a trace.                                                               |
| <b>Piracy</b>                | The unauthorized copying of a computer program written by someone else.                                                                                                                                |

|                      |                                                                                                                                                  |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Replication</b>   | The process of self-isolation within the current host program and attachment of the isolated segment to a new host.                              |
| <b>Robotics</b>      | The science that deals with robots, their construction, capabilities, and applications.                                                          |
| <b>Scores virus</b>  | Infects any application; loss of data due to system crashes.                                                                                     |
| <b>Time Bomb</b>     | A virus that checks for a specific date or time (such as Friday the 13th) before executing its objective.                                        |
| <b>Trap Door</b>     | A segment of code that allows data to be transmitted to the legitimate operator while other data are transferred to an illegal operator.         |
| <b>Trojan Horse</b>  | A programmer's placement of unauthorized instructions in a computer operating system for future use in gaining access to restricted information. |
| <b>Vaccine</b>       | Program that watches for typical things viruses do, halts them, and then warns the computer operator.                                            |
| <b>Viral strain</b>  | An original virus design; distinct from other viruses by its internal structure.                                                                 |
| <b>Viral variety</b> | A modification of a viral strain that results in a slightly different functionality.                                                             |
| <b>Virus remnant</b> | The remains of a virus (if any) after activation.                                                                                                |

## UNIT OVERVIEW

- I. **General issues**--as computers do more and more work in society, new social issues and questions continue to emerge. Many questions arose during the period when large mainframes took over data management in banks, insurance companies, financial institutions, and other large corporations; others appeared with the proliferation of microcomputers. Some of the effects of computers are felt by all of us.

A. Dependency

1. Anyone who has had to wait for several minutes because "the computer is down" while trying to make a credit card purchase at a retail store knows what it means to be dependent on computers.
2. Not learning the status of a credit card account is minor compared to the failure of the police or fire department emergency computer system or of an airline reservation system.
3. In the situations above, the manual substitute, or backup, is totally unsatisfactory, and these functions can be said to be "computer dependent."
4. There are dozens of ways in which our daily activities could be crippled by computer failure.

B. Nameless society

1. Can we accept being identified by a number instead of a name?
  - a. We are known to some organizations by our social security number, to others by our drivers license number, and to others by an account number.
  - b. If we respond to a computerized billing error, we have no assurance that a human will ever see the response.
2. It is unclear what effects this namelessness will have on the quality of our lives or where these effects will lead.

C. Technological developments

1. **Artificial intelligence**--a field of study involving techniques under which computers can formulate responses on the basis of content of accumulated and experience-related information.
  - a. Used in expert systems.
  - b. Used in decision support systems.



- c. Used in robotics.
- 2. **Robotics**--the science that deals with robots, their construction, capabilities, and applications.
  - a. Useful for repetitive work on an assembly line.
  - b. Used as mail carriers in large firms.
  - c. Used in warehousing.
  - d. Used by the handicapped.
  - e. Used by those working with radioactive materials.
- 3. **CD-ROM**--the current compact disks for computers, similar to read-only memory; they cannot be erased or recorded on.
  - a. About the same size as a floppy disk.
  - b. Store and retrieve faster than floppy disks.
  - c. Can hold a gigabyte (a million megabytes) or more.
  - d. Used in libraries.
  - e. Used for storing archival materials.
- 4. **Hypergraphics**--a method of instruction that involves the participants, the teacher, and the computer through the use of a response pad and an interconnection box.
  - a. Used in classrooms.
  - b. Used as a training device.
- D. **Viral epidemic**
  - 1. **Computer virus**--a segment of machine-language code that replaces software or data with a copy of itself, without authorization.
  - 2. **Pakistani Brain virus** (IBM and compatibles)
    - a. Replaces original boot sector with itself.
    - b. Moves original boot sector to another location.
    - c. Adds seven sectors which contain the remainder of the virus.



- d. Flags all modified sectors as unusable to protect itself.
  - e. Replicates onto all inserted bootable floppies.
- 3. **Scores virus (Macintosh)**
  - a. Infects any application.
  - b. Increases application size by 7000 bytes.
  - c. Seeks out new hosts at 3 1/2-minute intervals.
  - d. Creates notepad and scrapbook files in system folder.
  - e. Creates invisible Scores and Desktop files.
  - f. Looks for existence of specific file names for destruction.
- 4. **Israeli virus (IBM and compatibles)**
  - a. Infects any .COM or .EXE program.
  - b. Increases program's size by approximately 1800 bytes.
  - c. Infected programs are modified to become memory resident (TSRs).
  - d. Programs are infected when they are executed in the infected system.
  - e. Floppy or hard disks can become infected.
- 5. **nVIR virus (Macintosh)**
  - a. This virus appears in many different varieties.
  - b. Each variety has its own individual activation characteristics.
  - c. The publication of the source code for this virus is primarily responsible for the wide variations.
  - d. The similar infection techniques are:
    - (1) Places nVIR resource in system file; Code resource in application.
    - (2) Once system is infected, every application executed is infected.

6. **Alameda virus** (IBM and compatibles)
  - a. Replaces original boot sector with itself.
  - b. Stores original boot sector on first free sector.
  - c. Infects through software re-boot sequence.
  - d. Does not flag original boot sector as unusable.
7. **Lehigh virus** (IBM and compatibles)
  - a. Infects COMMAND.COM file.
  - b. Changes size by approximately 20 bytes.
  - c. Changes creation date and time.
  - d. Activates after four infections.
  - e. Destroys all system data.
8. **Datcrime virus**
  - a. Breaks down the directory section.
  - b. Makes it impossible for a disk's contents to be read.
9. **Logic Bomb virus**--checks for a certain action; for example, to see if an infected program has been executed a preset number of times.
10. **Network worms**
  - a. Attacks systems that are linked via communication lines.
  - b. May be found in programs that use network connections to spread from system to system.
  - c. Behaves like a virus or it can plant a virus.
11. **Time Bomb virus**--checks for a specific date or time (such as Friday the 13th) before executing its objective.
12. **Trojan Horse**--a programmer's placement of unauthorized instructions in a computer operating system for future use in gaining access to restricted information.

13. **Trap Door**--a segment of code that allows data to be transmitted to the legitimate operator while other data are transferred to an illegal operator.
14. **Peace virus**--a type of virus that was created to spread a greeting of world peace and then erased its own instructions, disappearing without a trace.

## II. Legal and ethical issues

### A. Computer crime

1. Any illegal act for which knowledge of computer technology is used to commit the offense.
2. A computer crime does not mean just breaking a written law; it also means breaking ethical codes.
3. Breaking ethical codes is a crime that is very hard to prosecute because of the way the laws are written.
  - a. Big corporations do not always prosecute because they want the crime "hushed up" so it will not hurt their business.
  - b. Corporations do not want to risk losing their stockholders; therefore, most computer crimes are settled out of court.
4. More computer crime exists today because the criminals are being hired as computer consultants and specialists; no precedent has been set to use as an example to deter other potential criminals.

### B. Data spying

1. A form of illegal access to data.
2. Most notable are the cases where unauthorized persons learn passwords or otherwise bypass security procedures in order to gain information from a competitor or for personal gain.
3. These people normally "look" at the data; they do not alter or delete data.
4. Data spying is most easily done when systems have teleprocessing capabilities to acquire or change information.
5. A personal computer or terminal can be used to dial up a computer. Industry must go to great lengths to improve password and log-on procedures in order to prevent this kind of activity.

C. Hackers

1. Computer enthusiasts who are challenged by the practice of breaking computer security measures designed to prevent unauthorized access to a particular computer system.
2. While only a few of these individuals actually engage in unauthorized or illegal activities, these few give a bad name to the whole class of gifted experimenters.
3. Rather than steal data, a hacker is more inclined to destroy or alter it.

D. Employees or former employees

1. Personnel who have access to confidential data are subject to temptations to reveal or sell information.
2. Disgruntled or fired employees have been known to sell or give away data for revenge or, when in a position to do so, remove or alter important records from the files.
3. Sometimes when an employee changes jobs, the knowledge about confidential information acquired in his/her former job may put him/her in a possibly awkward position.

E. Piracy

1. The unauthorized copying of a computer program written by someone else.
2. This crime has become "big business" in both the U.S. and Europe.
3. Software programmers are not putting the copy protection on their programs, which makes pirating easy.

F. Laws involving computer and software usage

1. The courts have ruled that computer programs are protected under the copyright laws.
2. The problem is that programs like audio and videotapes are easy to copy, making the law difficult to enforce.
3. Some publishers have gone to great lengths to insert special routines or sometimes viruses into their software to prevent illegal copying. NO method has been foolproof, and programs to break the copy protection routines are as common as the protection routines themselves.

4. Other publishers rely on the honor system, hoping that legitimate users will appreciate the trust which permits them to protect their own interests by making archival backups.
5. Currently the trend among major software companies is to offer unprotected versions of their copyrighted software programs.
  - a. Many companies sell site licensing contracts to businesses and schools.
  - b. Other companies just enclose a copyright page inside the software package container.

### III. Privacy issues

- A. When business was transacted using handwritten or typed documents, there was a high degree of control over the destination of copies of the information. Now, since computers process the data once it is entered, the control is less certain.
- B. The originator of the data, a person who filled out an employment application, for example, has little idea where the information will end up. Confidential data is processed by many organizations.
- C. Credit bureaus
  1. The sole purpose of these organizations is to keep up-to-date records on the financial activities of individuals and businesses.
  2. Data are compiled from member or affiliate businesses who subscribe to their services.
  3. A problem for the person holding the account is that inaccurate information can be difficult to remove from these credit bureau records.
  4. Sometimes it is easier for creditors to get information than it is for the individual concerned.
- D. Data banks
  1. There was once a proposal for a huge national data bank in which all the information about every citizen, gathered by all state and local agencies, would be stored.
  2. Using a social security number, any agency could have access to everything known about the owner of the number.
  3. The idea would save huge amounts of money because it would prevent much duplication of effort and would make information so much more accessible.

4. Fortunately, several members of Congress foresaw the plan as the serious threat to privacy that it was, and the data bank was never created. There are, however, many data bases within businesses and state and local governments that have a great deal of confidential data about persons. Improper access to these databases could be damaging.

E. Paperless offices

1. This term implies that all paper documents have been replaced by electronic media.
2. Office personnel may object to a "paperless office" because memos, reports, etc., would no longer be personal and confidential.
3. Security and passwords would have to be controlled so that sensitive information could be sent on the system.
4. Encryption
  - a. A term describing data that has been translated into a secret code for security reasons.
  - b. People who send E-mail messages to a specific person may use encryption.
  - c. Encryption could be used to keep computer crime to a minimum.
5. Some companies are using passwords to enter the computer systems.

IV. **Reliability of systems**--when speaking of computer reliability, most people think only of the hardware. Systems, however, are made up of hardware, software, and people. Reliability, or lack thereof, can be a factor of any of these elements.

A. Hardware

1. Although computer breakdowns are not as frequent as the public perceives them to be, hardware breakdowns do occur.
2. With mainframes, most organizations have a plan for backup operation of critical systems. The expense for the backup is usually justified by the importance and timeliness of the data.
3. Minicomputers are smaller than mainframes; however, the minis may have the same capabilities as a mainframe. Most companies will have some form of backup in case of failure.
4. When failures occur with microcomputers, the work can usually be transferred to another computer.



5. No matter what kind of a computer is used, there should always be a backup plan so that a hardware problem does not cripple the business operation.

B. Software

1. There is a saying that states, "There is no such thing as a program without bugs, only programs for which all the bugs have not yet been discovered."
2. Reputable software companies are responsive to users who have problems, and they often provide subscription support services to users.
3. As with storage, frequent backup copying is the best defense against software failure.

C. Storage media

1. Magnetic tape and disk packs are still being used on mainframes and minicomputers.
2. Diskettes, or fixed disks, are used for microcomputer storage.
3. Making backup copies of important data should be a regular activity, with the frequency of copying depending on the difficulty of recovering lost data.
  - a. When using magnetic tape or disk packs, they should be checked at regular intervals for wear and flaws.
  - b. When fixed (hard) disks are used, the copies should not be on the same disk.
  - c. Floppy disks should be inspected regularly for wear or flaws.

D. People

1. In large computer shops, several people are trained to do each job, and substitutes can step in when needed.
2. In an office using microcomputers, often only one person is assigned to a job, and if that person is ill or leaves, there is chaos while someone else learns the system.
3. Having formally written, precise, and detailed procedures for all jobs is the solution to this problem, but too often these procedures are not put in writing.

E. Confidentiality

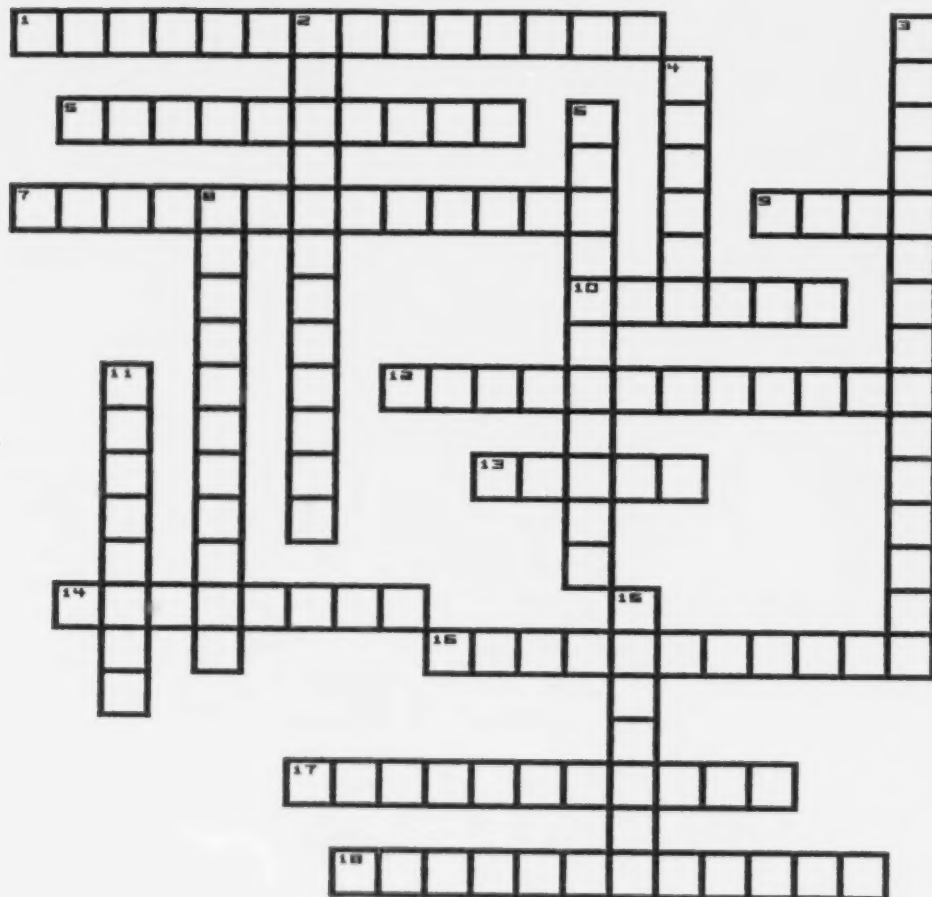
1. Computer professionals at all levels come in contact with private or even secret data.
2. Just as medical personnel and lawyers must preserve the confidentiality of their patients' or clients' records, a data entry operator must not divulge data that passes across his/her desk or terminal.
3. After leaving one job for another, or even when fired, former employees should not divulge knowledge about computer information.
4. Computer information must remain the property of the employer.

F. Accuracy

1. Loss of data can occur due to many causes--anything from industrial sabotage to hardware failure.
2. The most frequent cause of erroneous data is careless behavior by employees. Inaccurate key entry, failure to follow procedures for verification, leaving a workstation unattended, and overlooking an item of data are all ways in which an employee can be responsible for errors.
3. Improper care of equipment and disks can cause serious problems, including loss of data.
4. Not following specified exiting procedures may cause files to be inaccessible.



### ACTIVITY 1: TERMINOLOGY CROSSWORD PUZZLE



## ACTIVITY 1: CROSSWORD PUZZLE CLUES

### ACROSS CLUES

1. A term used to refer to the standard of moral conduct in computer use; a way in which the "spirit" of some laws are applied to computer-related activities.
5. A term describing data that has been translated into a secret code for security reasons.
7. A method used in classrooms or in-service training that involves the participants, the teacher, and the computer through the use of a response pad and an interconnection box.
9. Appears in many different varieties, each with its own individual activation characteristics. Causes loss of data and programs; causes frequent system crashes.
10. The unauthorized copying of a computer program written by someone else.
12. Programs that use network connections to spread from system to system; can behave like a virus, or it can plant a virus.
13. The current compact disks for computers, similar to read-only memory; they cannot be erased or recorded on.
14. The science that deals with robots, their construction, capabilities, and applications.
16. A programmer's placement of unauthorized instructions in a computer operating system, for future use in gaining access to restricted information.

17. Infects any application; loss of data due to system crashes.
18. The remains of a virus (if any) after activation.

### DOWN CLUES

2. A program that mimics the decision-making logic of human experts.
3. A term that implies that all paper documents have been replaced by electronic media and devices.
4. A term used to describe computer enthusiasts who are challenged by the practice of breaking computer security measures designed to prevent unauthorized access to a particular computer system.
6. The program to which the virus attaches itself.
8. The process of self-isolation within the current host program and attachment of the isolated segment to a new host.
11. A virus that checks for a specific date or time (such as Friday the 13th) before executing its objective.
15. Program that watches for typical things viruses do, halts them, and then warns the computer operator.

## ACTIVITY 2: ISSUES

### Project 1

**INSTRUCTIONS:** As our society becomes more dependent on computers and we become a "nameless society," many problems arise. Using the knowledge you have about this unit, write a one-page essay about how computer dependency and the nameless society affect you.



## ACTIVITY 2 (Continued)

### Project 3

**INSTRUCTIONS:** Define and give an example of how each of the following could be used in an office setting.

1. Robotics
2. Artificial Intelligence
3. CD-ROM
4. Hypergraphics

### ACTIVITY 3: LEGAL AND ETHICAL ISSUES

#### Project 1

**INSTRUCTIONS:** Write a short paragraph concerning the action you would take if you were in the following situations.

##### Situation 1

A friend just bought the last copy of a computer game that you have wanted for a long time. You have not been able to afford the game. Your friend tells you that since there is no copy protection on the diskette, he/she will make a copy and give it to you. What would you do?

##### Situation 2

Your school has just purchased a spreadsheet package for the computer lab. You have a home computer and one of your classes requires a lot of work done with a spreadsheet. The copy in the lab is not copy protected. Your best friend is the lab assistant and he/she tells you that if you want a copy of the spreadsheet package, you can copy it from one of the terminals. What would you do?

### ACTIVITY 3 (Continued)

#### Situation 3

You have a home computer and your brother/sister likes to "play" with the computer. One Saturday afternoon, he/she tells you that he/she has broken into someone's computer bank. What would you do?

#### Project 2

**INSTRUCTIONS:** Even though both federal and state laws are enacted toward stopping computer crime, many of the laws are not enforced. Of the following, research two (2) and explain what the laws are, what the laws cover, and what your opinion would be to improve the laws.

1. *Fair Credit Reporting Act of 1970*
2. *Freedom of Information Act of 1970*
3. *Federal Privacy Act of 1974*
4. *Counterfeit Access Device and Computer Fraud and Abuse Act of 1984*
5. *Small Business Computer and Security Education Act of 1984*
6. *Title 18 U.S. Code, Section 1029*
7. *Foreign Corrupt Practices Act of 1977*
8. *Texas Penal Code, Section 33.01-33.05*
9. Other laws suggested by the instructor

### ACTIVITY 3 (Continued)

#### Project 3

**INSTRUCTIONS:** Using the different types of viruses described in this unit, choose three (3) and explain what each one does to a computer system.

1)

2)

3)



#### ACTIVITY 4: PRIVACY ISSUES

**INSTRUCTIONS:** Research the following areas, and then write a report about what could be done to improve security (use terms discussed in the unit such as E-mail, etc.).

1. Credit bureaus
2. Data banks
3. Paperless offices

## ACTIVITY 5: RELIABILITY OF SYSTEMS

**INSTRUCTIONS:** Complete the following questions.

1. Name at least five (5) different types of hardware.
  - 1)
  - 2)
  - 3)
  - 4)
  - 5)
2. What can be done to protect software from failure?
3. Name three (3) types of data storage.
  - 1)
  - 2)
  - 3)
4. Write a short paragraph about the relationship involving people, confidentiality, and accuracy.

## SECTION I: INTRODUCTION TO BUSINESS COMPUTER PROGRAMMING

### UNIT III: COMPUTER-RELATED CONCEPTS

#### UNIT

#### OBJECTIVE:

Students will demonstrate the ability to understand computer processing concepts.

#### STUDENT

**COMPETENCIES:** After completing this unit, the student will be able to:

1. Identify basic computer-related terminology.
2. Outline the functions of data processing.
3. Illustrate the parts of the information processing cycle and explain how each part is interdependent within the cycle.
4. Illustrate parts of the computer processor.
5. Compare the similarities and differences among various computer systems.
6. Outline the events in history pertinent to breakthroughs in data processing.
7. Describe and categorize characteristics of computer languages.

#### FOCUS:

Knowledge of general computer operation and background is necessary in order to understand the more complex computer programs.

1. The ability to identify basic computer-related terms is essential in the day-to-day use of computer-related concepts.
2. Understanding each part of the information processing cycle and how the parts work together is imperative. The student can then process data and, at the same time, understand how the data were processed.

BCPI/Introduction to Business Computer Programming  
Unit 3: Computer-Related Concepts  
Student's Guide/OBJECTIVES

3. Identification of various computer systems is important in choosing the system most suitable for the task at hand.
4. Understanding the history of data processing is necessary to understand the development of computers.
5. Categorization and definition of computer languages allows the student to decide which language is most effective in specific problem solving.

## TERMINOLOGY

|                            |                                                                                                                                                                                                                                                                                               |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Analog</b>              | Computers which measure continuously and are designed to translate physical conditions such as temperatures, pressure and voltage in large quantities. Used mainly by scientists and engineers.                                                                                               |
| <b>Assembly language</b>   | Developed in the early 1950s to make using machine language easier. Replaced the binary code with symbolic names that were easier to remember.                                                                                                                                                |
| <b>Auxiliary storage</b>   | Generally magnetic tapes or magnetic disks on which data can be stored outside of the computer. Secondary storage also refers to auxiliary storage.                                                                                                                                           |
| <b>ASCII</b>               | American Standard Code for Information Interchange. ASCII is the most commonly used coding scheme today. It has a maximum character set of 128 codes; 0 through 127. Used mostly with microcomputers.                                                                                         |
| <b>Backup copies</b>       | Copies made of programs and data in case the original is damaged or destroyed.                                                                                                                                                                                                                |
| <b>Batch processing</b>    | The accumulation and processing of data as a group. Payrolls for large companies are usually completed through batch processing.                                                                                                                                                              |
| <b>BCD</b>                 | Binary-Coded Decimal. This was designed to encode numeric information in a manner that would be easier than straight binary to convert into decimal. Each decimal digit in a decimal value is translated into its binary equivalent and is recorded in BCD position which contains four bits. |
| <b>Character</b>           | A single letter, number, special symbol. One character equals one byte.                                                                                                                                                                                                                       |
| <b>Compiler</b>            | A program that interprets computer statements written in a symbolic form and converts the statements to machine language instructions.                                                                                                                                                        |
| <b>CPU</b>                 | Central Processing Unit is where all processing in the computer takes place.                                                                                                                                                                                                                  |
| <b>Data communications</b> | Transmission of data from one computer to another.                                                                                                                                                                                                                                            |
| <b>Data processing</b>     | The handling or manipulating of information in order to achieve a desired result.                                                                                                                                                                                                             |
| <b>Data structure</b>      | The arrangement of data for storing.                                                                                                                                                                                                                                                          |

|                            |                                                                                                                                                                                                              |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Detail file</b>         | A file that contains a record of day-to-day transactions.                                                                                                                                                    |
| <b>Digital</b>             | A computer that works by numbers and follows a set of instructions. Example--the computers used in the classroom are digital.                                                                                |
| <b>Direct access</b>       | Data can be stored and retrieved in any order. Also known as random access.                                                                                                                                  |
| <b>EBCDIC</b>              | Extended Binary-Coded Decimal Interchange Code. It is an eight-bit code that accommodates as many as 256 different characters. Usually the standard coding scheme for mini and mainframe computers.          |
| <b>Field</b>               | One piece of information, such as a name, address, or date.                                                                                                                                                  |
| <b>File</b>                | A group of related records treated as a unit.                                                                                                                                                                |
| <b>Hardware</b>            | Everything you can touch; just what the name implies, the physical parts associated with the computer. <i>Peripheral</i> is another term which refers to input and output devices connected to the computer. |
| <b>High-level language</b> | A computer language that uses English-like commands.                                                                                                                                                         |
| <b>Input</b>               | The raw data that need to be processed.                                                                                                                                                                      |
| <b>K</b>                   | A measurement of a computer's memory, 1024 bytes.                                                                                                                                                            |
| <b>Machine language</b>    | A computer language expressed in binary notation.                                                                                                                                                            |
| <b>Master file</b>         | A file that contains relatively permanent records.                                                                                                                                                           |
| <b>Menu</b>                | A list of options provided to the user by the computer program.                                                                                                                                              |
| <b>Modem</b>               | A device used in data communications to connect computers through phone lines.                                                                                                                               |
| <b>Output</b>              | Communicating the results of the processing of the user.                                                                                                                                                     |
| <b>Processing</b>          | When the facts are manipulated to produce results.                                                                                                                                                           |
| <b>Program</b>             | A series of instructions which directs the computer to perform a sequence of tasks that produces a desired output.                                                                                           |
| <b>Programmer</b>          | Person who designs, writes, tests, and implements the programs which process data.                                                                                                                           |
| <b>Record</b>              | A group of related fields.                                                                                                                                                                                   |

**BCPI/Introduction to Business Computer Programming**  
**Unit 3: Computer-Related Concepts**  
**Student's Guide/TERMINOLOGY**

|                               |                                                                                                                                                                     |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Sequential access</b>      | Data retrieved from a file one record after another in a predetermined sequence.                                                                                    |
| <b>Software</b>               | All instructions and information necessary to make the computer function. Computer programs used to accomplish specific applications are considered to be software. |
| <b>Transaction processing</b> | Processing data one record at a time. Making airline reservations is done as transaction processing.                                                                |
| <b>User</b>                   | The person using the computer.                                                                                                                                      |

## UNIT OVERVIEW

- I. **Definition of data processing**--the handling or manipulating of information in order to achieve a desired result.
- II. **Functions of data processing**
  - A. **Data collecting**--collecting raw data that need to be processed.
  - B. **Recording**--putting data in some form that is recognizable by person or machine.
  - C. **Sorting**--putting data in ascending/descending alphabetical or numerical order.
  - D. **Classifying**--arranging data according to some logical relationship.
  - E. **Calculating**--performing mathematical operations.
  - F. **Storing and retrieving**--storing data on some type of media for future use.
  - G. **Summarizing**--condensing data and information.
  - H. **Communicating**--transferring data from one place to another.
- III. **Information processing cycle** consists of four steps: input, processing, output, and storage. After data have been processed, they must be stored for further use. All functions of data processing are used in the information processing cycle.
  - A. **Input**--the raw data that need to be processed. Before data can be processed, they must be entered into the computer. An example of raw data is three scores that need to be averaged: 80, 85, 90. Data are input into the computer by input devices. It is very important that the correct data are input in order for the processing to be done correctly. If the input is wrong, the output will be wrong.
  - B. **Processing**--when the facts are manipulated to produce results. The average of the three scores is the process:  $(80+85+90)/3$ . The data are processed in the processor. The processor is considered to be the brain of the computer and has five functions: (1) makes logical decisions; (2) makes arithmetic computations; (3) serves as temporary storage; (4) edits and moves data; and (5) controls input and output devices.
    1. **Central processing unit**--all processing in the computer goes through the CPU, which is divided into two parts.
      - a. **Arithmetic/logic unit**--allows comparisons to be made in order to make logical decisions.
      - b. **Control unit**--controls all operations of the processor.



2. **Main memory**--considered to be primary storage, it is used for current computations.
  - a. **Read-only memory (ROM)**--contains chips that store permanent instructions for the computer. The user has no access to this memory to erase it.
  - b. **Random-access memory (RAM)**--where programs in use are stored. This memory also stores data before they are recorded outside of the computer. An example is a report being typed using a word processing program. The report is stored in RAM until it is recorded on a disk. The memory is erased when the computer is turned off. If the program is to be used again, it must be reloaded into the main memory.
- C. **Output**--communicating the results of the processing to the user. Output devices deliver the results of the processing. The output given for the grade average would be 85. This could be displayed on the screen or perhaps printed on paper.
- D. **Auxiliary storage**--storage outside of the main storage area. Auxiliary storage is needed in all types of computer systems. It expands memory capacity and allows access to data.
  1. **Secondary storage media**
    - a. **Magnetic tape**--cassette tapes or tape drives are popular storage devices. The tape is oxide-coated mylar, and the data are stored as magnetic spots.
    - b. **Magnetic disk**--data are recorded as magnetized spots. These disks may be either floppy or hard, and they are available in several sizes. Information is recorded on and read from the disks by a read/write head located in the disk drive which senses the magnetic spots.
    - c. **Laser disk**--data are stored by burning holes in the surface with a laser beam.
    - d. **Bubble memory**--contains small magnetic bubbles on a thin, single-crystal film of synthetic garnet. The presence or absence of a bubble is "on" or "off."
  2. **Types of access to stored data**
    - a. **Sequential**--data are retrieved in the order they were stored, one record after another. This type of access is mainly used for batch processing. It is the only retrieval method that can be used for magnetic tape.

- b. **Direct**--also known as *random-access*. Records can be read in any order, regardless of how they were stored. Magnetic disks are direct-access devices.

IV. **Types of computers**--computer systems vary widely in how the hardware is designed. However, what the hardware does is similar in all systems. Size is the most basic difference among computer systems. Four computer systems are listed below.

A. **Supercomputers**--large computers with fast processing speeds that can access billions of characters of data.

1. The CRAY is one of the best known examples of supercomputers. The CRAY sells for about \$20 million. The services of two full-time engineers are included in the purchase price. Another example of a supercomputer is the CYBER.
2. Both are used in such applications as nuclear physics, meteorology, and petroleum engineering.
3. Installation requires a specially prepared subflooring to support the weight of the computer and special plumbing to carry the fluorocarbon fluid needed as coolant.
4. About 150 supercomputers have been installed through 1987.

B. **Mainframe**--computers built to minimize the distance between points for very fast operations. Used for extremely complicated computations.

1. Cost over \$400,000 and can exceed \$1,000,000.
2. Serve more than one user, because they support large networks of individual terminals and remote, job-entry locations.
3. Used by banks, large commercial and industrial companies, and government agencies. Most professional programmers work in a mainframe environment.

C. **Minicomputer**--moderate-sized computer. Used when a desktop computer is not powerful enough to do the job.

1. Can be relocated conveniently, because they have no special power or environmental control requirements.
2. Smaller, slower, and less expensive than mainframes, but perform the same tasks as mainframes (only on a reduced scale).
3. Cost between \$10,000 and \$40,000.
4. Store and retrieve data from the same type of input and output devices as mainframes do, but with a smaller storage capacity.

- D. **Microcomputer**--the smallest desktop computers that are largely owned by individuals. The name derives from its main computing component being located on one integrated circuit or chip.
  - 1. The slowest computer, but the least expensive.
  - 2. Limited in the volume of data it can process.
  - 3. Fits on a desk or table.
  - 4. Does not require a data processing expert to operate it.
  - 5. Supports peripheral devices.
- E. **Laptop**--a small portable computer weighing nine pounds or less.
- V. The history and development of data processing actually has early roots. Men and women tried to develop tools to aid in calculating answers to problems and keeping track of the results. However, it was not until after the 18th century, a period known as the Industrial Revolution, that great strides were made. There were many events, discoveries, and inventions that culminated in our present Computer Age.
  - A. The following devices were employed or developed between ancient times and 1800.
    - 1. **Fingers**--digits (fingers and toes) were considered to be the first calculating device. Early Roman schools taught finger counting.
    - 2. **Abacus**--this was the first counting machine, and it was used over 3,000 years ago by the Chinese. It was made up of rods and beads.
    - 3. **Napier's bones**--John Napier invented a device consisting of rods or bones to perform calculations in 1614.
    - 4. **Pascaline**--in 1642, Blaise Pascal, a Frenchman, invented the first mechanical adding machine which could add and subtract. The machine used gears and wheels.
    - 5. **Leibnitz wheel**--in 1671, Gottfried von Leibnitz, a German, developed a calculator that could add, subtract, multiply, divide, and do square root. The device did not work very well because of inadequate technology at that time.
  - B. The following people and/or devices contributed between 1800 and 1900.
    - 1. **Jacquard's loom**--in 1804, Joseph Jacquard, a Frenchman, developed the loom which could automatically weave patterns on cloth. The automatic looms were operated by paper cards with holes punched in them.

2. **Analytical Engine**--in 1833, Charles Babbage, an Englishman, designed a machine which included a memory, calculator, and a control part which moved numbers back and forth between memory and calculator. The machine was designed to use punched cards as input and output. The machine was never built, because again the technology was not advanced enough to build the parts.
3. **Ada Augusta Lovelace**--Countess Lovelace, daughter of Lord Byron, collaborated with Charles Babbage on the analytical engine. She is referred to as the first programmer because of her concept of the loop--a set of instructions repeated frequently.
4. **Dr. Herman Hollerith**--in the 1880s, Dr. Hollerith developed the punched card mechanical processing machine. It was used by the Bureau of the Census for the 1890 census which took only 2 1/2 years. He organized a company to manufacture and market his machines; this company later became known as the International Business Machines Corporation (IBM).

C. The following people and devices contributed to the evolution of the computer from 1900 to the present.

1. **T. J. Watson, Sr.**--in 1924, T. J. Watson bought Hollerith's company and changed the name to IBM to convey the broadness of the scope he wanted for his company. He expected absolute loyalty and lifetime commitment from his employees and ran the company like a football coach/cheerleader. IBM financed the Mark I in 1944.
2. **Mark I**--this device was developed by Howard Aiken in 1944. This mechanical device was the first machine to use a stored program concept. It used punched cards as input and output. It was controlled automatically by instructions punched into paper tape attached to it. It was not an electronic machine.
3. **Grace Murray Hopper**--this navy commander began her career working with computers while a programmer on the Mark I in 1944. She coined the term "bug" and "debug." Grace Hopper also is known as the "mother of COBOL" because of her contributions to the writing of the programming language.
4. **ENIAC**--in 1946, the first electronic calculator was developed by Dr. John W. Mauchly and J. Presper Eckert under a contract from the U.S. Army. This machine contained thousands and thousands of vacuum tubes.
5. **EDSAC and EDVAC**--these electronic computers were developed to perform mathematical operations by stored program and without human intervention. These computers marked the beginning of what is called "The Computer Age."

6. **UNIVAC I**--this computer was manufactured by Sperry Rand in 1951, and it was the first commercial computer. General Electric and Metropolitan Life Insurance were the first companies to use the UNIVAC I for business purposes. The cost was \$1.5 million, and it contained 500 vacuum tubes.
7. **Charles Tandy**--in 1963, this entrepreneur bought Radio Shack, a supplier of hobby supplies. Tandy turned its focus to electronics. He marketed a low-cost personal computer for electronic hobbyists and then modified the Model II for business applications. He also pioneered the personal computer sales with the TRS-80 by using retail outlets, a new marketing technique at that time. His company continues to influence the computer market with its headquarters in Ft. Worth, Texas.
8. **H. Ross Perot**--this entrepreneur created the largest computer service firm in the world, Electronic Data Systems (EDS). He began his career at IBM as a computer salesman in Dallas. Perot sold EDS to General Motors in 1984. He continues to be active in other computer ventures, as well as educational, philanthropic, and political activities.
9. **Jack Kilby**--this engineer invented the integrated circuit/silicon chip/microchip in 1958 while working at Texas Instruments in Dallas, Texas. He determined that all the parts of a circuit--resistors, capacitors, and transistors--could be made of silicon and manufactured in a single block. He shares the patent with Robert Noyce.
10. **John Kemeny/Thomas Kurtz**--these Dartmouth professors of mathematics wanted to make computers easier so that everyone could use them. They developed the first BASIC program in 1964 on a time-sharing system. Other educators quickly adapted this programming language for their classes at all educational levels on microcomputers.
11. **Nolan Bushnell**--in 1972, this entrepreneur began his company, Atari, which is a term used in the Japanese game "go" that means "I'm going to attack you." He was the first person to make a video game that fit into a box that could be put in an arcade. He developed Pong, a form of electronic table tennis.
12. **Stephen Wozniak and Steve Jobs**--these two men collaborated in producing the original Apple computer in 1976. They manufactured the first Apple in the garage of Jobs' parents; it had one circuit board with 62 chips--no keyboard, case, memory, or power supply. Neither is presently associated with the company. Jobs developed the NeXt company in 1988 for applications in a college environment.
13. **Jack Tramiel**--in 1977, Jack Tramiel showed his first computer, the Commodore PET, at an electronics show in Chicago. He followed the PET with the VIC-20 and the Commodore 64. His computers were cheap. His comment was, "We sell to the masses, not the classes."



14. **Daniel Bricklin**--this entrepreneur developed the first electronic spreadsheet VisiCalc while in college taking business classes. It was written for the Apple II in BASIC. In 1979 VisiCalc was offered commercially for \$100 retail.
  15. **Adam Osborne**--in 1981, Adam Osborne introduced his Osborne computer. His computer was designed to fit under an airplane seat. He bundled the computer with software, for example WordStar and SuperCalc.
  16. **Mitch Kapor**--in 1983, his software program--Lotus 1-2-3--put the words "integrated package" on everybody's lips. This software was developed for the new IBM PC with a 16-bit processor. He chose the name Lotus, which is considered by the Hindus as a symbol of perfect enlightenment.
- D. **Computer generations**--the development of computers is divided into five generations according to dates. These generations are listed below by dates with a short description of the developments of each period.
1. **First**--from about 1952 to 1958, generation-one computers were characterized by their use of vacuum tubes. These computers were very slow, expensive to run and build, and unreliable. Heat and maintenance were also major problems. Applications were usually payroll processing and recordkeeping. The IBM 650 and UNIVAC I are examples of first-generation computers.
  2. **Second**--from about 1959 to 1964, second generation computers used transistors, which were smaller and required less power than vacuum tubes. Computers could now be programmed with English-like commands instead of numbers. Applications were batch-oriented, such as billing, payroll processing, and updating inventory files. The IBM 1401, Honeywell 200, and the CDC 1604 were second-generation computers.
  3. **Third**--from 1965 to 1970 came the introduction of the integrated circuit. Instead of large boards, circuits were developed on single chips of silicon. Third-generation computers were about 900 times faster than first-generation computers. IBM introduced its 360 series mainframe computer and the smaller, minicomputer.
  4. **Fourth**--microprocessors derived from integrated circuits put microcomputers on the office desk. Fourth-generation computers are roughly 100 times smaller than first-generation computers. These processors were introduced in 1971.
  5. **Fifth**--many computer magazines and books state that we are now entering the fifth-generation of computing. Fifth-generation computing is based on the advent of artificial intelligence operating systems. These operating systems use terminology very close to normal English syntax. Artificial intelligence systems are programmed

with enormous banks of data, based on expert opinion. When the user gives a command, the program will evaluate the command, and if more input is necessary from the user, it will ask for it.

VI. **Computer languages**--the computer operates under or is controlled by the direction of a program. The sequence of steps to be performed must be translated into detailed instructions the computer understands. Therefore, the programs must be written in a computer language understood by the computer being used. There are six categories of computer languages.

A. Categories of languages

1. **Level**--low-level languages are those which require the programmer to have detailed knowledge of the computer hardware system. A high-level language is one that is fairly machine-independent in that little or no knowledge of specific computing machinery is required.
2. **Purpose**--if a language can be used to solve a variety of problems, they are considered to be general-purpose. If the problems to be solved are specific, they are special-purpose languages.
3. **Structure**--if it is relatively easy to implement the structure mechanisms, the language is considered to be structured.
4. **Orientation**--when programmers indicate the procedures or algorithms in solving a problem, they use a procedure-oriented language. A programmer can specify what is to be accomplished in a problem-oriented language, leaving the development of procedures to the language.
5. **Translation**--every language must be translated into zeros and ones, except for machine language. Translation is performed by another computer program called a language translator. The language translator uses the program prepared by the programmer as input and produces the machine language as output.
6. **Interaction**--this means the programmer can conduct a dialogue with the computer--enter a line of code, have the computer test it, do an arithmetic computation, etc., in any order and at any time.

B. **Machine language**--a computer language expressed in binary notation. All other languages must be converted to machine language before the computer will understand them. Each computer has its own unique machine language written in binary code.

C. **Assembly language**--developed in the early 1950s to make using machine language easier. Assembly language replaced the binary code with symbolic names that were easier to remember.

- D. **High-level languages**--a computer language that uses English-like commands. Easier than machine language. These do not require the programmer to have working knowledge of the computer's internal components. Some popular computer languages are listed below.

1. **FORTRAN--FORmula TRANslator**

- a. This language was developed primarily for engineering, mathematical, and scientific applications.
- b. Characteristics of FORTRAN
  - (1) This language is the oldest high-level language.
  - (2) It can quickly execute mathematical problems.
  - (3) It is not well-suited for file maintenance, editing of data, or production of documents.
  - (4) This language is used primarily for solving scientific, engineering, and mathematical problems.
- c. Advantages of using FORTRAN
  - (1) Popular because it is widely used by programmers.
  - (2) Many available programs written in FORTRAN.
- d. Disadvantages of using FORTRAN--There is often a problem with compatibility among different versions, which causes a problem with transportability.

2. **COBOL--Common Business-Oriented Language**

- a. Developed in 1960 to be used in business applications.
- b. Dr. Grace Hopper, U.S. Navy, was a strong force behind the development of COBOL.
- c. Very popular, procedure-oriented language that is easy to understand, maintain, and modify.
- d. Used heavily in commercial and government computer installations.
- e. Advantages of using COBOL
  - (1) More like English than FORTRAN.
  - (2) Easy to read and tends to be self-documenting.



- (3) Reasonably machine-independent, so programs are transportable.
  - f. Disadvantages of using COBOL
    - (1) Requires a very large compiler.
    - (2) Coding is very slow, because there is so much to write.
- 3. **BASIC--Beginner's All-purpose Symbolic Instruction Code**
  - a. BASIC was developed at Dartmouth college to help students learn to program.
  - b. Advantages of using BASIC
    - (1) It is simple and flexible.
    - (2) It can be learned by people with no programming experience.
    - (3) It can be used for a wide variety of applications.
    - (4) It is primarily used on microcomputers.
    - (5) Easy for microcomputers to retrieve data from mainframe data files.
    - (6) It has good diagnostic messages.
  - c. Disadvantages of using BASIC
    - (1) Lack of standardization
    - (2) Limited transportability
    - (3) Slow execution speed
    - (4) Unstructured nature
- 4. **PASCAL**
  - a. Named for Blaise Pascal, a French inventor.
  - b. Experiencing the most rapid growth rate.
  - c. Used on microcomputers.
  - d. Should soon gain acceptance in the business world.
  - e. Advantages of using PASCAL

- (1) Well-suited for both batch and interactive uses.
  - (2) Easy to use, powerful, flexible, and self-documenting.
  - (3) Requires less memory.
  - (4) Easier to learn.
  - (5) Requires a computer which enables it to execute faster than other languages.
  - (6) Its detailed graphics capabilities make it useful to scientists and business people.
- f. Disadvantages of using PASCAL
  - (1) Lacks sophisticated input and output capabilities, which limits its use in business applications.
  - (2) Is not standardized, so transportability can be a problem.
- 5. **RPG--Report Program Generator**
  - a. Developed in about 1964.
  - b. RPG was designed to process data for generating and printing business reports.
  - c. Used for processing files for accounts receivable, accounts payable, inventory, and general ledger.
- 6. **C**
  - a. Developed by AT&T to shorten the time of system programming.
    - (1) Works on all computers.
    - (2) Used for system programs, utility programs, and graphics applications.
    - (3) May also be used for text processing and database programs.
    - (4) Has also been used to develop special effects in many films.

- b. Advantages of using C
  - (1) Application programs can be written in a matter of months rather than years.
  - (2) It is machine-independent and transportable, yet it provides the control of an assembler.
  - (3) C is user-oriented and machine-efficient.
  - (4) It is structured, allowing the programming to build complex programs out of simple parts.
  - (5) It is fast and requires little storage.
- c. Disadvantages of using C
  - (1) It is a difficult language to code.
  - (2) It is hard to read.

7. ADA

- a. Named in honor of Ada Lovelace.
- b. It was developed for use by the U.S. Defense Department in their search for a highly reliable language that could run on many different computers.
- c. Most recent and sophisticated of procedure-oriented languages.
- d. It produces the very accurate results needed in defense, weather forecasting, and oil exploration.

8. PL/1--Programming Language 1

- a. IBM's answer to languages with shortcomings such as FORTRAN and COBOL.
- b. It combines commands found both in COBOL and FORTRAN as well as its own unique commands.
- c. Problem-oriented language
  - (1) It is considered an extensive, sophisticated modular language.
  - (2) Its large compiler prevents the use of PL/1 on micros.
- d. Is used for both scientific and business applications.

9. **LOGO**

- a. LOGO was designed at MIT as a teaching tool.
- b. It enables adults and children to begin programming and communicating with the computer in a short time.
- c. It helps the inexperienced user learn logic and programming.

10. **APL--A Programming Language**

- a. Symbolic interactive language used primarily by engineers, mathematicians, and scientists.
- b. Provides programmers with a "shorthand" that speeds up the process of coding.
- c. Advantages of using APL
  - (1) A freedom style with a lack of restrictions on input and output.
  - (2) Has extremely powerful mathematical capabilities.
  - (3) Fast language with fewer restrictions on data than any other language.
  - (4) Handles tables of information easily which appeals to business application programmers.
- d. Disadvantages of using APL
  - (1) Very hard to read.
  - (2) Requires a special keyboard and printer.
  - (3) File handling capabilities are limited.
  - (4) Requires very large amounts of storage.
  - (5) Is not available on microcomputers.

11. **FORTH**

- a. Designed as a fourth-generation computer language.
  - (1) It is fairly low-level language developed for micro-computers in the 1970s.

- (2) The name *FORTH* was derived from *FOURTH*, because filenames were limited to five characters on the computer used to develop this language.

b. Advantages of FORTH

- (1) Excellent control of the computer, because it is close to assembly language.
- (2) Good for industrial process control applications.
- (3) Standard control language for astronomical observatories.
- (4) Used by many industry leaders, such as IBM and Hewlett-Packard, for in-house development.

c. Disadvantage--limited vocabulary functions.

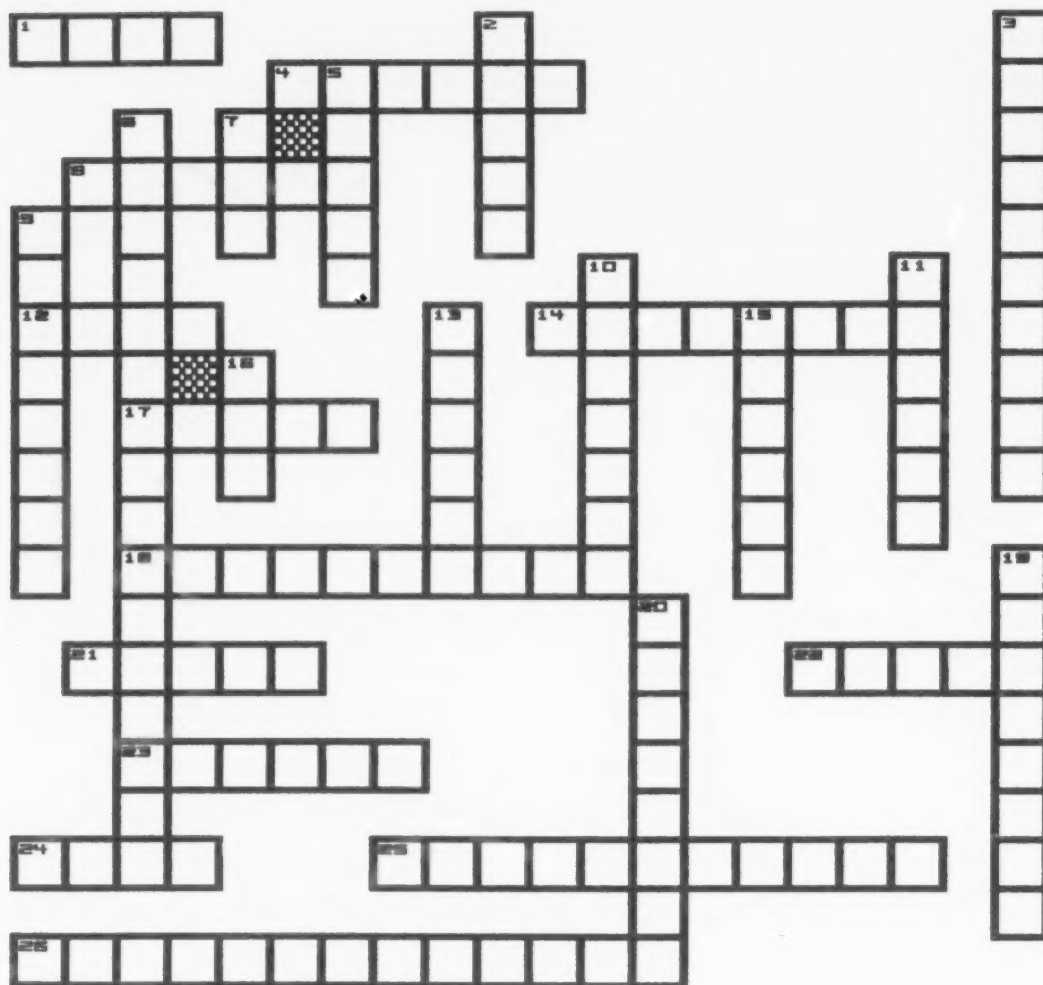
E. Selecting computer languages

1. Speed
2. Purpose of the program
3. Time and money required to design and code a program
4. Availability of languages on the system used
5. Storage requirements of the compiler
6. Language standardization and transportability
7. Potential equipment changes in the future

F. **Language translator**--high-level languages must be translated into machine language in order to be processed by the computer.

1. **Interpreter**--an interpreter takes each instruction written in a high-level language and translates that instruction into machine code.
2. **Compiler**--a compiler does the interpretation just once and transfers the high-level language into machine code. A compiled program runs much faster but cannot be changed or corrected after it is compiled.

## ACTIVITY 1: TERMINOLOGY CROSSWORD PUZZLE



## ACTIVITY 1: TERMINOLOGY CROSSWORD PUZZLE CLUES

### ACROSS

1. A list of options provided to the user by the computer program.
4. An eight-bit code that accommodates as many as 256 different characters. Primarily used on mini and mainframe computers.
8. Communicating the results of the processing to the user.
12. A group of related records treated as a unit.
14. A measurement of the computer's memory.
17. The most commonly used coding scheme.
18. Data that is stored one file after the other.
21. A device used in data communications to connect computers through phone lines.
22. A device used in data communications.
23. Used mainly by scientists and engineers.
24. The person using the computer.
25. The type of processing where data is processed one record at a time.
26. The arrangement of data for storing.

### DOWN

2. One piece of information, such as a name, address, or date.
3. The person who designs, writes, tests, and implements the programs which process data.
5. Processing data as a group.
6. A device on which data can be stored outside of the computer. Another name for secondary storage.
7. Where all processing in the computer takes place.
9. Computer programs used to accomplish specific applications.
10. Computers which work by numbers and follow a set of instructions.
11. A group of related fields.
13. Records can be read in order regardless of how they were stored. Also known as random-access.
15. Copies made of programs and data in case the original is damaged or destroyed.
16. Designed to encode numeric information in a manner that would be easier than straight binary to convert into decimal.
19. A program that interprets computer statements written in a symbolic form and converts the statements to machine language instructions.
20. The physical parts associated with the computer.



## ACTIVITY 2: FUNCTIONS OF DATA PROCESSING

**INSTRUCTIONS:** Describe the functions of data processing listed below.

1. Summarizing
2. Calculating
3. Data collecting
4. Sorting
5. Communicating
6. Recording
7. Storing and retrieving
8. Classifying

### ACTIVITY 3: PARTS OF THE INFORMATION PROCESSING CYCLE

#### Project 1

**INSTRUCTIONS:** Draw and label a diagram showing the information processing cycle and describe each step in the cycle.

### ACTIVITY 3 (Continued)

#### Project 2

**INSTRUCTIONS:** Design an information processing cycle for the jobs listed below.

1. Writing a research paper
2. Preparing a recipe
3. Averaging gas mileage on a trip
4. Calculating pay
5. Averaging grades
6. Hiring an employee

### ACTIVITY 3 (Continued)

#### Project 3

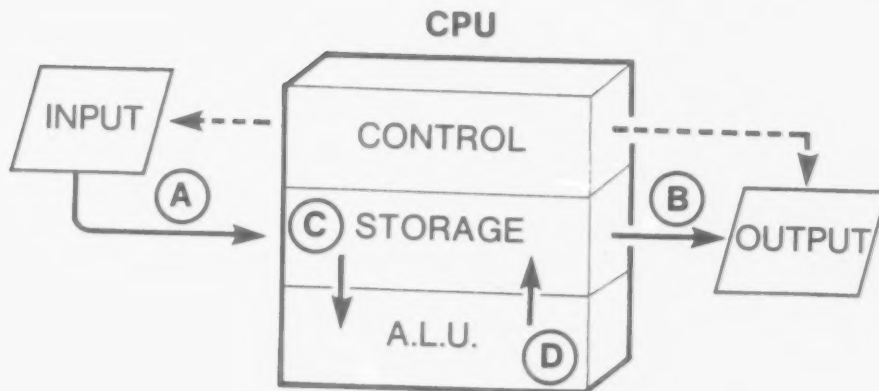
**INSTRUCTIONS:** Draw and label a diagram showing the parts of a processor. Illustrate the flow of data through the processor.

### ACTIVITY 3 (Continued)

#### Project 4

**INSTRUCTIONS:** Using the illustration below, indicate the direction of data flow for the six steps given by placing the letter representing the direction of the data flow in the space provided.

- |       |    |                                          |
|-------|----|------------------------------------------|
| _____ | 1. | Input two numbers into the computer.     |
| _____ | 2. | Multiply the two numbers.                |
| _____ | 3. | Store the product of the multiplication. |
| _____ | 4. | Divide the product by 4.                 |
| _____ | 5. | Store the division result.               |
| _____ | 6. | Display the result on the monitor.       |



## ACTIVITY 4: COMPUTER SYSTEMS CAPABILITIES AND LIMITATIONS

### Project 1

**INSTRUCTIONS:** Explain the capabilities and limitations of the following computer systems.

1. Supercomputer:
  
  
  
  
  
  
  
  
  
  
2. Mainframe:
  
  
  
  
  
  
  
  
  
  
3. Minicomputer:
  
  
  
  
  
  
  
  
  
  
4. Microcomputer:
  
  
  
  
  
  
  
  
  
  
5. Laptop:

## ACTIVITY 4 (Continued)

### Project 2

**INSTRUCTIONS:** Evaluate the capabilities and limitations of computer systems and correlate them to the business situations listed below.

| BUSINESS SITUATIONS                                | COMPUTER SYSTEMS |
|----------------------------------------------------|------------------|
| ____ 1. Fifteen thousand-student school district   | A. Supercomputer |
| ____ 2. Word processing operator                   | B. Mainframe     |
| ____ 3. Accounting office with five employees      | C. Minicomputer  |
| ____ 4. Smallest computer                          | D. Microcomputer |
| ____ 5. Medical secretary                          | E. Laptop        |
| ____ 6. Very small community hospital              |                  |
| ____ 7. Major airline                              |                  |
| ____ 8. Portable                                   |                  |
| ____ 9. Data processing center for a major factory |                  |
| ____ 10. Law firm with four lawyers                |                  |
| ____ 11. Elementary school teacher                 |                  |
| ____ 12. Nuclear physics                           |                  |



## ACTIVITY 4 (Continued)

### Project 3

**INSTRUCTIONS:** Interview six people who use computers on the job. Ask what type of system is used, and why. Take notes on each interview. Prepare a short summary comparing why various computers are used in different jobs.

Listed below are examples of questions which could be asked.

1. What type of computer system is used in this business?
2. What criteria were used in the decision to buy this system?
3. How many systems were compared before the purchase?
4. How much did the system cost?
5. What applications are used the most on the system?
6. How much time is the computer used daily?
7. How much training were the employees involved in when learning how to use the system?
8. How many different employees use the system?

### Project 4

**INSTRUCTIONS:** Look through a variety of computer and business magazines and select pictures of computer systems. Prepare a poster that identifies each computer system by the five types studied. If possible, include prices and descriptions.

## ACTIVITY 5: HISTORY OF DATA PROCESSING

### Project 1

**INSTRUCTIONS:** Match the dates shown below to the inventions that correspond to them.

| YEAR |            | INVENTION |                           |
|------|------------|-----------|---------------------------|
| A.   | 1614       | 1. _____  | Babbage's analytic engine |
| B.   | 1890       | 2. _____  | Microprocessor            |
| C.   | 1946       | 3. _____  | Jacquard's loom           |
| D.   | 1804       | 4. _____  | UNIVAC I                  |
| E.   | Late 1940s | 5. _____  | Leibnitz wheel            |
| F.   | 1833       | 6. _____  | Napier's bones            |
| G.   | 1971       | 7. _____  | IBM 360                   |
| H.   | 1642       | 8. _____  | EDSAC and EDVAC           |
| I.   | 1944       | 9. _____  | Hollerith's punched card  |
| J.   | 1671       | 10. _____ | Mark I                    |
| K.   | 1951       | 11. _____ | ENIAC                     |
| L.   | Late 1960s | 12. _____ | Pascaline                 |

## ACTIVITY 5 (Continued)

### Project 2

**INSTRUCTIONS:** Draw a time line illustrating the development of calculating machines and computers in the history of data processing, starting with fingers and ending with microprocessors. Include at least one characteristic to describe each device shown on the time line.

## ACTIVITY 5 (Continued)

### Project 3

**INSTRUCTIONS:** Prepare a one-page written report about one of the inventions discussed in data processing history, and explain how it aided in the development of computers.

### Project 4

**INSTRUCTIONS:** Select one of the people who made contributions to the computing industry and write a one-half page report. Make an oral presentation to the class.

## ACTIVITY 6: COMPUTER LANGUAGES

### Project 1

**INSTRUCTIONS:** Answer the following questions regarding computer languages using complete sentences.

1. List six ways to categorize computer languages.

1.

2.

3.

4.

5.

6.

## ACTIVITY 6 (Continued)

### Project 1 (Continued)

2. List the language or languages in which most business programs are coded.
3. Identify a language designed to process data to generate reports.
4. Name a new language used for system programs, utility programs, and graphic applications.
5. Uses a "shorthand" that speeds up the coding process.
6. Name the language Grace Hopper played a role in developing.
7. Identify the oldest high-level language.

## ACTIVITY 6 (Continued)

### Project 1 (Continued)

8. Name the primary language for microcomputers.
9. Low-level language developed for microcomputers in 1870s.
10. Name the language used by the U.S. Department of Defense.
11. Name the language which made machine language easier.

## ACTIVITY 6 (Continued)

### Project 2

**INSTRUCTIONS:** After comparing the characteristics of each computer language, select the language or languages that would best fit the jobs listed below.

- |       |                                              |
|-------|----------------------------------------------|
| _____ | 1. Business programmer                       |
| _____ | 2. Engineering                               |
| _____ | 3. High school student learning to program   |
| _____ | 4. Programmer for U.S. Department of Defense |
| _____ | 5. An employee preparing reports             |

### Project 3

**INSTRUCTIONS:** Describe what a high-level language is and list three positive characteristics.

A high-level language is:

Three positive characteristics:

- 1.
- 2.
- 3.



## SECTION II

### COMPUTER HARDWARE UTILIZATION

#### ESSENTIAL ELEMENTS

75.87(g)(2)(A)(B)(C)(D)

2. Concepts and skills related to hardware utilization. The student shall be provided opportunities to:
  - (A) Review computer hardware operations;
  - (B) Understand categorization of hardware according to size, application, and configurations;
  - (C) Reinforce the ability to use basic operating systems; and
  - (D) Review the proper care of hardware and software.

## SUGGESTED LESSON PLAN

### SECTION II: COMPUTER HARDWARE UTILIZATION

#### UNIT I: COMPUTER HARDWARE

|                                |                            |                            |                |
|--------------------------------|----------------------------|----------------------------|----------------|
| <b>ESSENTIAL<br/>ELEMENTS:</b> | <b>75.87<br/>(g)(2)(B)</b> | <b>ALLOCATED<br/>TIME:</b> | <b>5 Hours</b> |
|--------------------------------|----------------------------|----------------------------|----------------|

**UNIT  
OBJECTIVE:** Students will reinforce their knowledge of all phases of operating computer hardware.

**STUDENT  
COMPETENCIES:** Upon completion of this unit, the student will be able to:

1. Identify the types of computers.
2. Describe primary memory.
3. Describe various input devices.
4. Identify various output devices.
5. Identify other hardware devices.

**FOCUS:** Being familiar with all phases of computer hardware operation is valuable in most businesses.

**DOMAINS:** Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:** Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. Types of computers
  - A. Supercomputers
  - B. Mainframes
  - C. Minicomputers
  - D. Microcomputers
  - E. Portable computers
- II. Hardware components
  - A. Central processing unit
  - B. Primary memory
  - C. Secondary memory
    - 1. Hard disk
    - 2. Floppy disk
    - 3. Disk storage structure
    - 4. Size of disk
    - 5. File organization methods
  - D. Monitors/cathode ray tube
  - E. Input devices
    - 1. Keyboard
    - 2. Magnetic ink character recognition
    - 3. Optical character recognition
    - 4. Universal product code
    - 5. Magnetic input
    - 6. Tablets
    - 7. Mice

8. Touch screens
  9. Light pens
  10. Joy sticks
  11. Voice recognition
- F. Printers
1. Dot matrix
  2. 24-pin dot matrix
  3. Daisy wheel
  4. Laser
  5. Postscript laser
  6. Ink jet
  7. Chain and band
  8. Page
- G. Plotters
1. Drum
  2. Flatbed
- H. Modem
- I. Tape drives
- J. Internal hardware
- K. Optical disk
- L. Bernoulli boxes
- M. Hard cards

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

Specific Teaching Activities:

1. Obtain examples of:
  - a. Microfiche and a reader, if possible.
  - b. Optical scanning media.
  - c. Manuals of school's computer and peripheral devices.
  - d. Inventory form.
  - e. Catalogs from vendors and suppliers of computer merchandise.
  - f. External tapes.
  - g. Types of computer paper and forms.
2. Visit a computer center to observe:
  - a. An operator using console commands.
  - b. An operator reading a tape drive and disk drive.
  - c. Various types of printers.
  - d. An operator entering data via terminal/CRT operation of modems.
3. Tour a facility that uses optical scanning devices, and observe how unreadable and/or damaged documents/items are handled.

NOTE: The types of computers have been introduced in Section I Introduction to Business Computer Programming I, Unit III, Computer-Related Concepts. While some of the information is the same, it is greatly expanded upon in this unit.

## ACTIVITY MATRIX

[illegible]

1988



## COMPUTER SPEEDS

### MILLISECOND

$$\frac{1}{1,000} \text{ SECOND}$$

### MICROSECOND

$$\frac{1}{1,000,000} \text{ SECOND}$$

### NANOSECOND

$$\frac{1}{1,000,000,000} \text{ SECOND}$$

### PICOSECOND

$$\frac{1}{1,000,000,000,000} \text{ SECOND}$$

## SUPPLEMENTARY ACTIVITY

### COMPUTER BINGO

**NOTE:** This game can be used with any section or combination of sections in this unit to aid in teaching and reviewing new material.

**Materials needed:** 1 sheet of typing paper per student  
pencils  
computer terms

#### INSTRUCTIONS:

1. Give each student one sheet of paper.
2. Have students draw five vertical lines down the 11" side of the paper.
3. Have students draw four vertical lines down the 8 1/2" side of the paper. Twenty (20) squares should result.
4. Designate the second square on the third row as a free space.
5. Write a list of terms on the board. Have students select terms at random and place one term in each square on their paper.

#### HOW TO PLAY

1. The teacher reads definition for one of the terms.
2. The students who have the term being defined on their cards mark an X on that space on their cards.
3. The student who finds all terms on his/her card first, wins.

## COMPETENCY TEST

1. Define CPU.
2. List the parts of the CPU.
  - 1.
  - 2.
  - 3.
  - 4.
3. List and explain the two types of primary memory.
4. List five types of secondary storage devices.
  - 1.
  - 2.
  - 3.
  - 4.
  - 5.
5. List two input devices.
  - 1.
  - 2.

6. List and describe the three types of color monitors.

1.

2.

3.

7. List and discuss three types of printers.

1.

2.

3.

8. Compare the small, medium, and large computer systems using the following features:

CPU

Storage characteristics

Software

Peripherals

Expanded by adding

## ANSWERS TO ACTIVITIES

### Activity 1

All Projects Evaluated to the satisfaction of the instructor.

### Activity 2

Project 1 The following major components of the CPU should be included: bus, clock, registers, and arithmetic/logic unit.

Project 2 Evaluated to the satisfaction of the instructor.

### Project 3

1. CGA
2. VGA
3. EGA

Discussion of the monitors to the satisfaction of the instructor.

### Project 4

1. Modem
2. Plotters
3. Scanner
4. Mice
5. Tablets

Other peripherals are also acceptable. Descriptions of the peripherals to the satisfaction of the instructor.

### Project 5

1. Dot matrix
2. Daisy wheel--letter quality
3. Laser

Discussion of the printers to the satisfaction of the instructor.

## COMPETENCY TEST ANSWERS

1. CPU is also known as the Central Processing Unit. It is considered the brain of the computer.
2.
  1. Clock
  2. Bus

3. Register
4. Arithmetic/logic unit
3. Two types of primary memory are ROM and RAM. ROM stands for Read-Only Memory. ROM cannot be changed. It includes the general computer operations.  
  
RAM stands for Random-Access Memory. RAM is considered volatile memory, and it can be changed.
4.
  1. Diskette
  2. Hard disk
  3. Optical disk
  4. Bernoulli boxes
  5. Hard cards
5.
  1. Keyboards
  2. Mice
  3. Screens
6. CGA, VGA, EGA--described to the satisfaction of the instructor.
7. Laser, daisy wheel, dot matrix--others are acceptable. Described to the satisfaction of the instructor.
8. Described to the satisfaction of the instructor.

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(2)(B)

Students shall be provided the opportunity to understand categorization of hardware according to size, application, and configurations.

---

|                   |                                                                                              |                |                    |
|-------------------|----------------------------------------------------------------------------------------------|----------------|--------------------|
| <b>OBJECTIVE:</b> | <b>Students will reinforce their knowledge of all phases of operating computer hardware.</b> |                |                    |
|                   |                                                                                              | <b>MASTERY</b> | <b>NON-MASTERY</b> |

---

- |    |                                  |       |       |
|----|----------------------------------|-------|-------|
| 1. | Identify the types of computers. | _____ | _____ |
| 2. | Describe primary memory.         | _____ | _____ |
| 3. | Describe various input devices.  | _____ | _____ |
| 4. | Identify various output devices. | _____ | _____ |
| 5. | Identify other hardware devices. | _____ | _____ |
|    |                                  | _____ | _____ |
|    |                                  | _____ | _____ |
|    |                                  | _____ | _____ |

STUDENT NAME \_\_\_\_\_



## SUGGESTED UNIT LESSON PLAN

### SECTION II: COMPUTER HARDWARE UTILIZATION

#### UNIT II: COMPUTER SOFTWARE

|                               |                            |                                     |
|-------------------------------|----------------------------|-------------------------------------|
| <b>ESSENTIAL<br/>ELEMENT:</b> | <b>75.87<br/>(g)(2)(C)</b> | <b>ALLOCATED<br/>TIME: 12 Hours</b> |
|-------------------------------|----------------------------|-------------------------------------|

**UNIT  
OBJECTIVE:** Students will understand the role software plays in utilizing computer hardware.

**STUDENT  
COMPETENCIES:** After completion of this unit, the student will be able to:

1. Describe the characteristics and functions of an operating system.
2. Identify and categorize applications that would utilize batch processing and real-time processing.
3. Identify a compiler, interpreter utilities, and virtual storage.
4. Discuss the different versions of the microcomputer operating systems.
5. Identify the major components of the DOS/VS operating system.

**FOCUS:** A working knowledge of operating systems allows one to be more flexible with the type of computer he/she is able to operate, and this knowledge facilitates the worker in accomplishing basic computer tasks.

**DOMAINS:** Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:** Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. System software
  - A. Characteristics of operating systems
    - 1. IOCS
    - 2. Basic functions
    - 3. Overall function
    - 4. Types
  - B. Language translators
    - 1. Compilers
    - 2. Interpreters
    - 3. Assembler
  - C. Utilities
  - D. User interface shell
  - E. Virtual storage
- II. Types of operating systems
  - A. Microcomputer
    - 1. MS-DOS
    - 2. OS/2
    - 3. APPLE IIe DOS 3.3
    - 4. APPLE DOS 1.1
    - 5. MAC
  - B. Mainframe
    - 1. Control programs
    - 2. Operator to system communication
    - 3. Processing programs

- III. Application software
  - A. Word processing
  - B. Spreadsheets
  - C. Databases
  - D. Graphics
  - E. Communications
  - F. Integrated packages
  - G. Utilities
  - H. Programming languages

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

Specific Teaching Activities

1. Obtain examples of:  
Various mainframe JCL job sets.  
Manuals for the mainframe computer for the JCL.  
Manuals for the various DOS microcomputers.
2. Make available current literature on compilers, interpreters, and utilities for both mainframes and microcomputers.
3. The following Supplementary MS-DOS tests are available for use with this unit:  
MS-DOS COMMANDS TEST  
MS-DOS TEST  
OPERATING SYSTEMS SKILLS TEST  
Translate these to your computer system if you have MS-DOS equipment.
4. The following handouts are available for your use:  
Handout 1: FUNCTIONS OF DOS/VS OPERATOR  
Handout 2: EXAMPLES OF JOB COMMANDS  
Handout 3: TYPES OF OPERATOR COMMANDS

### ACTIVITY MATRIX

[illegible]





## **HANDOUT 1: FUNCTIONS OF DOS/VS OPERATOR COMMANDS**

By using the appropriate DOS/VS operator commands, you can perform the following functions:

### **SET SYSTEM VALUES**

The ADD command adds devices to the PUB table; the DEL command deletes devices from the PUB table. The CAT command identifies the disk drive on which the VSAM catalog is mounted. The SET command sets the values for date and time. The DPD command defines the page data set.

### **TEMPORARILY SUSPEND PROCESSING**

The PAUSE command causes a partition to pause between jobs or job steps, enabling you to take action such as mounting the next tape reel or disk pack.

### **CANCEL JOBS**

The CANCEL command terminates the execution of a job.

### **CHANGE I/O DEVICE ASSIGNMENTS**

The ASSGN command assigns a physical I/O device to a logical unit name. The DVCDN (device down) command informs the system that an I/O device is inoperative. The DVCUP (device up) command informs the system that an I/O device that was inoperative is now operational again. The RESET command resets temporary I/O assignments--which are valid for only one job--to the permanent assignments or, where no permanent assignments were made, to the standard assignments established when the system was generated.

### **CONTROL MAGNETIC TAPE OPERATIONS**

The MTC command controls magnetic tape operations, such as rewinding tapes, skipping files, and writing tape marks.

### **CLOSING FILES**

The CLOSE command closes a system or programmer logical unit assigned to magnetic tape, or a system logical unit assigned to a disk drive or a diskette input/output unit.

## OBTAIN INFORMATION FROM THE SYSTEM

The LISTIO command prints or displays a list of current I/O device assignments in the system. The LOG command prints or displays all job control statements and commands occurring in the partition in which the command is issued. The NOLOG command suppresses logging of the job control statements and commands.

## CONTROL MULTIPROGRAMMING

The following commands are valid only in a multiprogramming environment:

The ALLOC command enables you to allocate storage to virtual partitions; the ALLOCR command enables you to allocate storage to real partitions. The BATCH, START, UNBATCH, and STOP commands are used to initiate, interrupt, and reinitiate processing in a partition. The MAP command prints a map of the current sizes of the partitions, both real and virtual, as well as the size of the page pool. The PARTY (priority) command enables you to resume processing after a STOP command.

Although the normal communication device for operator commands is SYSLOG (assigned to a console printer-keyboard or display console), commands may also be entered from SYSRDR, with the exception of attention commands.

## HANDOUT 2: EXAMPLE OF A JOB

|                                                   |                                                                                                                                                                                                                                                                                                                                                                               |
|---------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| // JOB jobname                                    | The JOB statement identifies the job and indicates the beginning of the job control statements for that job.                                                                                                                                                                                                                                                                  |
| // ASSGN SYSLST,X'OOE'<br>// ASSGN SYS001,x'161'  | These are the ASSGN statements for the various job steps, assuming that these assignments differ from the current assignments. The assignments established by the ASSGN statements remain in effect until the /& statement, when they will be reset to the standard assignments (established during system generation), and/or to assignments specified by operator commands. |
| // OPTION LINK, DUMP,<br>LIST                     | The OPTION statement specifies that the output of the compiler and the assembler is to be written on SYSLNK if the job ends abnormally.                                                                                                                                                                                                                                       |
| // EXEC FCOBOL<br>(COBOL SOURCE DECK)<br>/*       | This EXEC statement specifies that the program, in this case the COBOL compiler, is to be executed. It must be followed by the COBOL source deck and by the /* (the end of data statement).                                                                                                                                                                                   |
| // EXEC LNKEDT                                    | This EXEC statement specifies execution of the linkage editor program, which then combines the COBOL object module on SYSLNK and places the link-edited program in the core-image library.                                                                                                                                                                                    |
| // EXEC (test data)<br>/* (end of data statement) | This EXEC statement causes the program to be executed. The blank operand indicates that the program to be executed is the one just link-edited. Then it is temporarily placed in the core-image library. The data for the program must follow this statement if SYSIPT and SYSRDR are assigned to the same device.                                                            |
| // PAUSE                                          | The PAUSE statement causes the job control program to stop processing temporarily so that you can act accordingly to the instructions contained in the statement. You may also enter commands at this time. Processing continues when you press END (or ENTER).                                                                                                               |

**/INCLUDE**  
**(object module name)**  
**/\* (end of data statement)**

The INCLUDE statement signals that an object module is to be included. Note that this is neither a job control statement nor an operator command.

(PHASE, INCLUDE, ENTRY, AND ACTION) If no operand is specified, the module to be included should follow on SYSIPT.

**/&**

The end-of-job statement causes all temporary I/O assignments to be reset to the standard assignment and/or to the assignments established by operator commands.

**// JOB Jobname**

Job statement for the next job.

## HANDOUT 3

### TYPES OF OPERATOR COMMANDS

#### IPL COMMANDS

IPL commands are accepted only at IPL time (when you perform the IPL procedure). These commands prepare the DOS/VS system for operation.

#### ATTENTION COMMANDS

Attention commands are accepted after you have pressed REQUEST on the console keyboard. (If 'AR' does not appear, press REQUEST again, after which the message 1140 REQUEST CANCEL [-ENTER POWER/VS COMMAND] is displayed.) Attention commands from a display console may be entered.

#### JOB CONTROL COMMANDS

Job control commands are only accepted between jobs or job steps.

#### POWER/VS COMMANDS

POWER/VS commands are treated like attention commands.

#### VTAM NETWORK OPERATOR COMMANDS

VTAM network operator commands are needed to monitor **and** control a VTAM telecommunication network.

## SUPPLEMENTARY TESTS

### MS-DOS COMMANDS TEST

**INSTRUCTIONS:** Give the appropriate MS-DOS command to accomplish the following requests.

(Note: Assume the current drive is A, unless the problem specifies differently.)

1. Information about the files on the disk in drive A.
2. Information about the files, that will give one screenful at a time, on the disk in drive B.
3. A list of the file names only on the disk in drive B.
4. Information about all BAS files on the disk in drive B, if drive B is the current drive.
5. Copy the file named HOOP.BAS, which is on a disk in drive A, to a disk in drive B.
6. Copy the file name PROG1.CBL, which is on a disk in drive A, to a disk in drive B, giving the new file the name of PROGRAM1.CBL.
7. Copy everything in the current directory on drive A to drive B.
8. Delete everything in the current directory on drive A with the extension of BAK.
9. Delete the file CHAPT5.BAK which is on the disk in drive B.

10. Rename the file \_\_\_\_\_ to \_\_\_\_\_. The file is on the disk in drive B, the current drive.
11. Display the content of AUTOEXEC.BAT, which is on the disk in drive A.
12. Format and put the system files on the disk in drive B.
13. Format the disk in drive B.
14. Format and put a volume label on the disk in drive A.
15. Check the disk in drive B for errors.
16. Make a subdirectory called WORD on the disk in drive A.
17. Change back to the root directory on the current drive.
18. Remove the empty subdirectory called WORK from the current drive.
19. Set the path to search drive A if drive B is the current drive.

## MS-DOS TEST

### Project 1

INSTRUCTIONS: Select the correct command for each description.

- |           |             |            |
|-----------|-------------|------------|
| A. A:     | J. DISKCOPY | S. PROMPT  |
| B. BACKUP | K. KEYBFR   | T. REN     |
| C. CD     | L. RD       | U. RESTORE |
| D. CHKDSK | M. EDLIN    | V. SORT    |
| E. CLS    | N. FDISK    | W. SYS     |
| F. COPY   | O. FIND     | X. TREE    |
| G. DATE   | P. FORMAT   | Y. TYPE    |
| H. DEL    | Q. MD       | Z. MORE    |
| I. DIR    | R. PRINT    |            |

- |                                                                |                                                              |
|----------------------------------------------------------------|--------------------------------------------------------------|
| _____ 1. Used to create a subdirectory on the disk.            | _____ 11. Changes from one subdirectory to another.          |
| _____ 2. Gives the number of files and bytes free on a disk.   | _____ 12. Removes a subdirectory from the disk.              |
| _____ 3. Erases a file from a disk.                            | _____ 13. Makes an exact copy of a file onto another disk.   |
| _____ 4. Displays the contents of a file on the screen.        | _____ 14. Makes an exact copy of one disk onto another disk. |
| _____ 5. Displays the contents of a file on the printer.       | _____ 15. Initializes a disk for use.                        |
| _____ 6. Allows the user to change the name of a file on disk. | _____ 16. A filter that allows you to order information.     |
| _____ 7. Displays the names of the files stored on a disk.     | _____ 17. A filter used to display one screen at a time.     |
| _____ 8. Erases the screen.                                    | _____ 18. A simple line-editor word processor.               |
| _____ 9. Used to enter the correct data into the computer.     | _____ 19. Displays all directories and subdirectories.       |
| _____ 10. Allows the user to customize a DOS prompt.           |                                                              |



Project 2

**INSTRUCTIONS:** Write out brief answers to the following questions.

1. Describe two reasons for the use of extension names.
  - a.
  - b.
2. What is meant by an internal command?
3. What is meant by an external command?
4. Why is there a distinction between internal and external commands?
5. What is the purpose of the AUTOEXEC.BAT file?
6. What is the purpose of a batch file?

7. What are four characteristics of an operating system?
  - a.
  - b.
  - c.
  - d.
8. What are four categories of system software?
  - a.
  - b.
  - c.
  - d.
9. What happens when you format a disk?
10. List two good reasons for using subdirectories.
  - a.
  - b.
11. What are the two wild card symbols and their functions?
  - a.
  - b.

Project 3

**INSTRUCTIONS:** Assume that MS-DOS is in drive A and that the data disk is in drive B. What are the commands to accomplish the following?

1. Assume you are in drive B. What sequence of commands must you type to see a directory of drive A on a clear screen?
2. Assume you are in the root directory on drive A. What sequence of commands must you enter to copy a file named FRIEND.DAT from this directory to the subdirectory WORDS on drive B?
3. What keys must you press to do a soft reboot of the system?
4. Make a duplicate MS-DOS disk.
5. Add all files from drive A to B.
6. Get rid of all files that begin with a W.
7. Change the extension name of all files with the extension .DOC to .TXT.
8. Create a bootable disk.
9. Redirect the DIRectory output to the printer.
10. Display the contents of \_\_\_\_\_ document to the screen.
11. Display the contents of \_\_\_\_\_ document to the printer.

## OPERATING SYSTEMS SKILLS TEST

**INSTRUCTIONS:** Assume MS-DOS is in drive A and your data disk is in drive B for all applicable questions. List the commands to accomplish the following:

1. Create a bootable data disk.
2. Put a volume label on a disk that has been formatted.
3. Put the system on a formatted disk.
4. Create a subdirectory for word processing, and within that subdirectory, create a subdirectory for storing documents.
5. From the root directory, change to the DOC subdirectory.
6. Copy a file from drive A to the DOC subdirectory.
7. Make a duplicate copy of a disk.
8. Copy all files from drive A to drive B.
9. Delete all files that start with F.
10. List all BASIC programs.
11. Echo the last DOS command.

**BCPI/Computer Hardware Utilization**  
**Unit 2: Computer Software**  
**Teacher's Guide/SUPPLEMENTARY TESTS**

12. Echo the last DOS command one character at a time.
13. Echo the last DOS command up to a specific letter.
14. Echo the last DOS command from a specific letter on.
15. Print the screen.
16. Rename a file DEMO.TXT.
17. Delete all files with the extension .SYS.
18. Clear the screen.
19. Remove the subdirectory DOC.
20. Copy a file from drive A to B and give it a new name.
21. Copy a file to the same disk with a new name.
22. Return to the root directory from a subdirectory.
23. Return to the next upper level subdirectory, and you do not know the name of that subdirectory.

**BCPI/Computer Hardware Utilization**  
**Unit 2: Computer Software**  
**Teacher's Guide/SUPPLEMENTARY TESTS**

24. Display the current VOLUME label.
25. Display all directories and subdirectories.
26. Display the current version of DOS.
27. Change the current date.
28. Reboot the computer.
29. Display the contents of a text file on the screen.
30. Display the contents of a text file on the printer.
31. Copy a listing of the directory to disk.
32. Copy a listing of the directory to the printer.
33. Sort all files on disk by file name.
34. Sort all files on disk by extension name.
35. Create a batch file from the console.
36. Display the amount of free space on drive B.

37. Change the prompt to display the current directory.
38. All files on your data disk have your initials for an extension name. You recently changed your last name and now you must change your initials on all the files.
39. Create a batch file that will be executed automatically when you boot the computer.
40. Create a batch file that will automatically start a program called WRITE in the subdirectory WORD when you type in the command TYPE.
41. Set up your data disk so that when you boot the computer, it will clear the screen and display a custom prompt showing the time and date on one line, the current directory on the next line, and followed by the > sign.
42. Search a disk for the file named \_\_\_\_\_.
43. Create a virtual (RAM) disk drive C with 100K of memory.

## COMPETENCY TEST

### PART I: CHARACTERISTICS OF OPERATING SYSTEMS

1. Define an operating system.
  
  
  
  
  
  
  
  
  
  
2. List an operating system's three basic functions.
  - 1.
  
  
  
  
  
  
  
  
  
  
  - 2.
  
  
  
  
  
  
  
  
  
  
  - 3.



## PART II: BATCH AND REAL-TIME PROCESSING

1. List three batch processing applications.

- 1.

- 2.

- 3.

2. Define batch processing.

3. Define multiprogramming.

4. Define real-time processing.

**PART III: COMPILERS, INTERPRETERS, UTILITIES,  
AND VIRTUAL STORAGE**

1. List five compilers and/or interpreters.
  - 1.
  - 2.
  - 3.
  - 4.
  - 5.
2. Define a compiler.
3. Define an interpreter.
4. Define utility programs and describe what they do.

**PART III (Continued)**

5. Define virtual storage.
  
  
  
  
  
  
  
  
  
  
6. List the two areas of virtual storage.
  - 1.
  - 2.

#### PART IV: MICROCOMPUTER OPERATING SYSTEMS

1. What is the name of the operating system that you use?
  
2. What happens in the initializing operation?
  
  
  
  
  
  
  
  
  
  
3. According to your operating system's documentation, what are the internal commands available? What are the external commands?

Internal commands

External commands

## PART V: COMPONENTS OF DOS/VS OPERATING SYSTEMS

1. List the three control programs of a mainframe.
  - 1.
  - 2.
  - 3.
2. List three of the five types of operator commands.
  - 1.
  - 2.
  - 3.
3. Which DOS/VS library contains program phases that have been processed by the linkage editor and are, therefore, ready to be executed?
4. Which DOS/VS library contains object modules, the output from language translators?

## ANSWERS TO ACTIVITIES

### Activity 1

1. The Input/Output Control System, or IOCS, might be considered to be the pioneer operating system. Its control commands are stored in an on-line library. The I/O control initiates, executes, monitors, and controls the transfer of data into and out of the computer. I/O control eliminates details of handling.

Answers may vary.

- 2.1 Job management is responsible for managing the total computer system. Communication between the computer and the operator is provided as a part of the job-management function.
- 2.2 System resource management allows the resources of the system to be allocated to programs being executed. These resources include the computer's memory, computational ability, and I/O devices.
- 2.3 Data management is creating, changing, and reorganizing files.
3. The overall function of the operating system is to control the activities of the computer system. It serves as the traffic cop, directing and managing computer events.

Activity 2    1.-5.    Evaluated to the satisfaction of the instructor.

### Activity 3

1. Answers may vary.  
Compilers are translators that translate the entire source code into object code.  
Interpreters translate the source code one line at a time as it is being executed.
- 2.-3. Evaluated to the satisfaction of the instructor.
- 4.1 The real address area covers all the addresses that exist in the real storage of the computer.
- 4.2 The virtual address area starts just beyond the highest address of the real address area and ranges up to a value that was specified when the DOS/VS system was generated. Virtual storage is generally considered to be larger than real storage.

- 5.-6. Evaluated to the satisfaction of the instructor.

#### Activity 4

1. MS-DOS 1.0 1981  
Version 1.1  
Version 2.0 1982  
Version 2.1  
Version 3.0 1984  
Version 3.2 1985  
Version 3.3 1987  
Version 4.0 1988

- 2.-5. Evaluated to the satisfaction of the instructor.

#### Activity 5

- 1.1 The initial program loader (IPL), which starts the system.
- 1.2 The supervisor, which coordinates the use of the system resources and maintains the flow of operations.
- 1.3 The job control program, which is needed for every program execution.
- 2.1 IPL commands
- 2.2 Attention commands
- 2.3 Job control commands
- 2.4 POWER/VS commands
- 2.5 VTAM network operator commands
- 3.1 Language translators
- 3.2 Service and system utility programs
- 3.3 Problem/application programs
- 4.1 Displaying the contents of a library
- 4.2 Adding elements (phases, modules, etc.) to a library
- 4.3 Deleting elements from a library
- 4.4 Condensing a library
- 4.5 Creating a library and utility programs
- 5.1 Core image
- 5.2 Relocatable
- 5.3 Source statement
- 5.4 Procedure
- 6.-7. Evaluated to the satisfaction of the instructor.

## MS-DOS COMMANDS TEST

### ANSWERS

1. A> DIR
2. A> DIR B: /P
3. A> DIR B: /W
4. B> DIR \*.BAS
5. A> COPY HOOP.BAS B:
6. A> COPY PROG1.CBL B:PROGRAM1.CBL
7. A> COPY \*.\* B:
8. A> DEL \*.BAK
9. A> DEL B:CHAPT5.BAK
10. B> RENAME BUDGET01.WKS BUDGET10.WKS
11. A> TYPE AUTOEXEC.BAT
12. A> FORMAT B:/S
13. A> FORMAT B:
14. A> FORMAT /V
15. A> CHKDSK B:
16. A> MKDIR WORD                      A> MD WORD
17. >CD\
18. >RMDIR WORK                      >RD WORK
19. B> PATH A:\



## MS-DOS TEST

### ANSWERS

#### Project 1

- |    |   |     |   |     |   |
|----|---|-----|---|-----|---|
| 1. | Q | 7.  | I | 13. | F |
| 2. | D | 8.  | E | 14. | J |
| 3. | H | 9.  | G | 15. | P |
| 4. | Y | 10. | S | 16. | V |
| 5. | R | 11. | C | 17. | Z |
| 6. | T | 12. | L | 18. | M |
|    |   |     |   | 19. | X |

#### Project 2

1.
  - a. Computer--distinguish file types, to know if they are executable, commands, batch, or other.
  - b. Human--distinguish files by types and for organizing files.
2. A DOS command that is part of the operating system. It is loaded when you boot the computer.
3. A DOS command that remains on disk. It is loaded into memory only when it is needed, thus saving primary memory space.
4. When MS-DOS was first created, memory was "prime real estate." All the operating system commands could not fit in primary memory. Therefore, the less used commands were left out of the operating system to conserve primary memory.
5. The AUTOEXEC.BAT file is a batch file that will automatically execute when the computer is first booted.
6. A batch file is a list of DOS and special commands, stored in a file, and are executed as if they were typed in from the keyboard.
7.
  - a. Communication between primary memory and peripheral devices
  - b. Method of transferring data from RAM to disk and from disk to RAM
  - c. Data management
  - d. Commands that allow you to manipulate the computer
8.
  - a. Operating system
  - b. Utilities
  - c. Language processors
  - d. User interface shell

9.
  - a. Erases anything that was on the disk.
  - b. Sets up tracks and sectors.
  - c. Creates the boot area.
  - d. Creates the FAT area.
  - e. Creates the root directory area.
  - f. Blocks out any bad sectors.
10.
  - a. To organize files into different directories, keep all similar files in individual directories.
  - b. Limitation of the root directory necessitates the use of subdirectories.
11.
  - a. ?--replaces single character.
  - b. \*--replaces 1 to 8 characters.

### Project 3

1. B>A:  
A>CLS  
A>DIR /W
2. A>COPY FRIEND.DAT B:\WORDS\
3. CTRL-ALT-DELETE
4. A>DISKCOPY A: B:
5. A>COPY \*.\* B:
6. A>DEL W\*.\*
7. REN \*.DOC \*.TXT
8. A>FORMAT B:/S
9. A>DIR > PRN
10. A> \_\_\_\_\_.DOC
11. A>PRINT \_\_\_\_\_.DOC or A>TYPE \_\_\_\_\_.DOC > PRN

## OPERATING SYSTEMS SKILLS TEST

### ANSWERS

- |                                       |                                                          |
|---------------------------------------|----------------------------------------------------------|
| 1. A>FORMAT B:/S                      | 19. A>CD \WORD\DOC change<br>to DOC subdirectory         |
| 2. A>LABEL B:                         | A>DEL *.* delete all files in<br>DOC directory           |
| 3. A>SYS B:                           | A>CD \WORD change to<br>WORD directory                   |
| 4. A>MD WORD<br>A>CD WORD<br>A>MD DOC | A>RD DOC remove DOC<br>directory                         |
| 5. A>CD \WORD\DOC                     | 20. A>COPY filename.ext<br>B:newname.ext                 |
| 6. A>COPY filename.ext<br>\WORD\DOC   | 21. A>COPY filename.ext<br>B:newname.ext                 |
| 7. A>DISKCOPY A: B:                   | 22. A>CD\                                                |
| 8. A>COPY *.* B:                      | 23. A>CD.                                                |
| 9. A>DEL F*.*                         | 24. A>TREE                                               |
| 10. A>DIR *.BAS                       | 25. A>VOL                                                |
| 11. F3                                | 26. A>VER                                                |
| 12. F1, or right arrow key            | 27. A>DATE                                               |
| 13. F2, then specified character      | 28. CTRL-ALT-DELETE                                      |
| 14. F4, then beginning character      | 29. A>TYPE filename.ext                                  |
| 15. Shift-PRTSCN                      | 30. A>PRINT filename.ext<br>or A>TYPE filename.ext > PRN |
| 16. A>REN DEMO.TXT<br>DEMO1.TXT       | 31. A>DIR > filename.ext                                 |
| 17. DEL *.SYS                         | 32. A>DIR > PRN                                          |
| 18. CLS                               | 33. A>SORT   DIR                                         |
|                                       | 34. A>SORT   DIR /+8                                     |

BCPI/Computer Hardware Utilization  
Unit 2: Computer Software  
Teacher's Guide/ANSWERS

- 35. A>COPY CON filename.BAT
- 36. A>CHKDSK B:
- 37. A>PROMPT \$P\$G
- 38. A>REN \*.INT \*.INS
- 39. A>COPY CON AUTOEXEC.BAT
- 40. A>COPY CON TYPE.BAT  
ECHO OFF  
CD \WORD  
WRITE  
^Z
- 41. A>COPY CON AUTOEXEC.BAT  
ECHO OFF  
CLS  
PROMPT \$T \$D\$\_\$P\$G
- 42. A>DIR MORE.COM
- 43. A>COPY CON CONFIG.SYS  
DEVICE=VDISK.SYS,100

## COMPETENCY TEST ANSWERS

### Part I

1. The operating system controls the activities of the computer system. It serves as the traffic cop, directing, and managing the computer events.
2. Evaluated to the satisfaction of the instructor.

### Part II

1. Evaluated to the satisfaction of the instructor.
2. Batch processing jobs are collected into groups for processing.
3. Multiprogramming allows the computer to run two or more jobs at the same time, sharing the computer's resources.
4. Real-time processing occurs immediately, and the results are available to affect the activity at hand.

### Part III

1. Evaluated to the satisfaction of the instructor.
2. Compilers are translators that translate the entire source code into object code.
3. An interpreter translates the source code one line at a time as it is being executed.
4. Utility programs perform common tasks such as routine file conversions and sort/merge operations.
5. Virtual storage is a condition in which a portion of the program is stored on disk and is brought into memory only as needed.
- 6.1 The real address area
- 6.2 The virtual address area

Part IV 1.-3. Evaluated to the satisfaction of the instructor.

### Part V

- 1.1 IPL
- 1.2 Supervisor
- 1.3 Job control
- 2.1 IPL

**BCPL/Computer Hardware Utilization**  
**Unit 2: Computer Software**  
**Teacher's Guide/ANSWERS**

- 2.2 Attention
- 2.3 Job control
- 2.4 POWER/VS
- 2.5 VTAM
- 3. Core image
- 4. Relocatable

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(2)(C)

The student will be provided opportunities to reinforce the ability to use basic operating systems.

---

---

**OBJECTIVE:** Students will understand the role software plays in utilizing computer hardware.

**MASTERY**      **NON-MASTERY**

---

---

1. Describe the characteristics and functions of an operating system.
2. Identify and categorize applications that would utilize batch processing and real-time processing.
3. Identify a compiler, interpreter utilities, and virtual storage.
4. Discuss the different versions of the microcomputer operating systems.
5. Identify the major components of the DOS/VS operating system.

|       |       |
|-------|-------|
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |
| _____ | _____ |

STUDENT NAME \_\_\_\_\_

## SUGGESTED LESSON PLAN

### SECTION II: COMPUTER HARDWARE UTILIZATION

### UNIT III: COMPUTER OPERATION AND MAINTENANCE

|                               |                               |                                     |
|-------------------------------|-------------------------------|-------------------------------------|
| <b>ESSENTIAL<br/>ELEMENT:</b> | <b>75.87<br/>(g)(2)(A)(D)</b> | <b>ALLOCATED<br/>TIME: 12 Hours</b> |
|-------------------------------|-------------------------------|-------------------------------------|

**UNIT  
OBJECTIVE:** The students' knowledge will be reinforced in all phases of operating computer hardware.

**STUDENT  
COMPETENCIES:** Upon completion of this unit, the student will be able to:

1. Identify terminology related to computer operation and maintenance.
2. Operate and maintain machines.
3. Monitor and report environmental problems.
4. Inventory resources and report findings.
5. Handle media and support tools.
6. Monitor hardware performance.
7. Schedule workflow.
8. Maintain data and physical security.

**FOCUS:** Being familiar with all phases of computer hardware operation is valuable in most businesses.

**DOMAINS:** Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.



**BASIC SKILLS:** Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. Operating and maintaining machines
  - A. Operating a mainframe system
  - B. Operator-to-system communication
  - C. Tape drive
  - D. Disk drives
  - E. Printers
  - F. CRT/Terminals
  - G. Optical scanning devices
  - H. Modems
  - I. Microfiche
- II. Monitor and report environment problems
  - A. Environment variables
  - B. Hazards
- III. Inventory resources and report the findings
  - A. Preparing forms
  - B. Inventory resources
  - C. Organizing storage
  - D. Purchasing
- IV. Handle media and support tools
  - A. Storage methods
  - B. Diskettes
  - C. Libraries
  - D. Backups

**BCPI/Computer Hardware Utilization**  
**Unit 3: Computer Operation and Maintenance**  
**Teacher's Guide/CONTENT OUTLINE**

- V. Monitor hardware performance
  - A. Error message manual
  - B. Classification of wait states
  - C. Operator commands
  - D. Interrupt processing
- VI. Schedule workflow
  - A. Operator run manual
  - B. Types of error messages
- VII. Maintain data and physical security
  - A. Access to data
  - B. Disaster and recovery plans
  - C. Backup and recovery
  - D. Copyright and ethics

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

**Specific Teaching Activities:**

1. Obtain examples of:
  - a. Microfiche and a reader, if possible.
  - b. Optical scanning media.
  - c. Manuals of school's computer and peripheral devices.
  - d. Inventory form.
  - e. Catalogs from vendors and suppliers of computer merchandise.
  - f. External tapes.
  - g. Types of computer paper and forms.
2. Visit a computer center to observe:
  - a. An operator using console commands.
  - b. An operator reading a tape drive and disk drive.
  - c. Various types of printers.
  - d. An operator entering data via terminal/CRT operation of modems.
3. Tour a facility that uses optical scanning devices, and observe how unreadable and/or damaged documents/items are handled.
4. Visit a business establishment, secure a speaker, or provide a film to describe the different types of fire extinguishing systems.

### ACTIVITY MATRIX

| Activities  | Domains   |              |           | Basic Skills |         |      |
|-------------|-----------|--------------|-----------|--------------|---------|------|
|             | Cognitive | Psycho-motor | Affective | Reading      | Writing | Math |
| 1           | X         |              |           | X            |         |      |
| 2           | X         |              | X         |              | X       |      |
| 3 Project 1 | X         |              | X         |              | X       |      |
| Project 2   | X         |              | X         | X            | X       |      |
| Project 3   | X         |              |           |              |         | X    |
| Project 4   | X         |              | X         |              |         | X    |
| Project 5   | X         | X            |           |              | X       | X    |
| Project 6   | X         |              |           |              |         | X    |
| 4 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         |              |           |              | X       |      |
| Project 3   | X         |              |           |              | X       |      |
| Project 4   | X         |              |           |              | X       |      |
| Project 5   | X         |              |           |              | X       | X    |
| 5 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         | X            |           |              | X       |      |
| Project 3   | X         |              |           |              | X       |      |
| Project 4   | X         |              |           |              | X       |      |
| Project 5   | X         | X            |           |              | X       |      |
| Project 6   | X         |              |           | X            | X       |      |
| Project 7   | X         |              |           |              | X       |      |
| 6           | X         |              |           | X            |         |      |
| 7 Project 1 | X         |              |           | X            | X       |      |
| Project 2   | X         |              |           |              | X       |      |
|             |           |              |           |              |         |      |
|             |           |              |           |              |         |      |

**BCPI/Computer Hardware Utilization**  
**Unit 3: Computer Operation and Maintenance**  
**Teacher's Guide/ACTIVITY MATRIX**

### ACTIVITY MATRIX

[illegible]

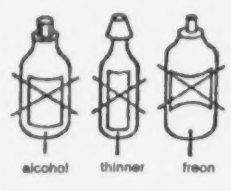
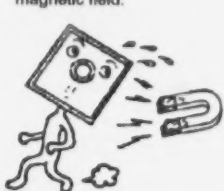
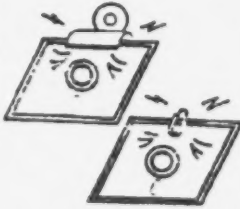
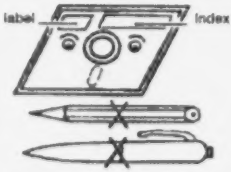
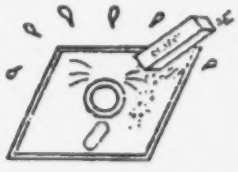
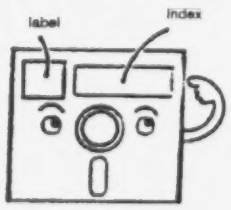
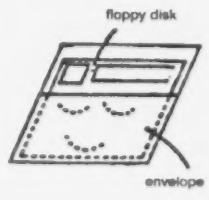
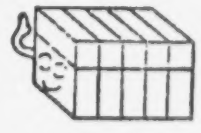
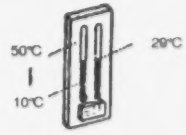
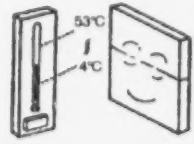
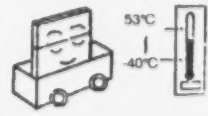
## COMPETENCY TEST MATRIX

[illegible]



## HANDOUT: FLOPPY DISK HANDLING AND STORAGE

### HANDLING PRECAUTIONS TO PROTECT AGAINST POSSIBLE FAILURE

|                                                                                                                                                                            |                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>1. Do not touch surface. It is easily contaminated which causes errors.</p>            | <p>2. Do not use alcohol, thinners or freon to clean disk.</p>                                                                                                              | <p>3. Do not use magnets or magnetized objects near the disk. Data can be lost from a disk exposed to a magnetic field.</p>  | <p>4. Do not bend or fold the disk.</p>                                                                                                                                                                       |
| <p>5. Do not place heavy objects on the disk.</p>                                        | <p>6. Do not use rubber bands or paper clips on the disk.</p>                                                                                                              | <p>7. Do not write on disk label with pencil or ball-point pen. Use felt-tip pen only.</p>                                  | <p>8. Do not use erasers on disk.</p>                                                                                                                                                                        |
| <p>9. Apply index label to the right of Maxell label. Do not use labels in layers.</p>  | <p>10. Insert carefully by grasping upper edge and placing it into the drive.</p>                                                                                         | <p>11. Keep disk in its protective envelope when not in use.</p>                                                           | <p>12. Disks not being used should be stored vertically in their box.</p>                                                                                                                                   |
| <p>13. Do not expose the disk to excessive heat or sunlight.</p>                        | <p>14. Operating environment:<br/>A. Temperature: 10° C to 50° C (50° F to 122° F)<br/>B. Wet Bulb Temperature: Less than 29° C<br/>C. Relative Humidity: 20% to 80%</p>  | <p>15. Storage environment:<br/>A. Temperature: 4° C to 53° C (40° F to 127° F)<br/>B. Relative Humidity: 8% to 80%</p>     | <p>16. While in transit the disk should be in its envelope in a protective box within the following ranges:<br/>A. Temperature: -40° C to 53° C (-40° F to 127° F)<br/>B. Relative Humidity: 8% to 90%</p>  |

## COMPETENCY TEST

### PART I: TERMINOLOGY

**INSTRUCTIONS:** Fill in the correct term next to the appropriate definition.

- |       |                                                                                                                         |
|-------|-------------------------------------------------------------------------------------------------------------------------|
| _____ | 1. Consists of the CPU, I/O devices, and the operating system.                                                          |
| _____ | 2. A specialized language for setting up jobs and communicating basic instructions to the computer.                     |
| _____ | 3. Contains a CPU on a single chip.                                                                                     |
| _____ | 4. Software that controls the execution of computer programs.                                                           |
| _____ | 5. A temporary storage device used for special functions.                                                               |
| _____ | 6. An organized method for achieving a given result.                                                                    |
| _____ | 7. An individual program, or routine, incorporated into a job.                                                          |
| _____ | 8. The amount of data that, within a given period, can be entered into the system, processed, and output in final form. |
| _____ | 9. Frequently used programs that perform tasks such as copying files.                                                   |
| _____ | 10. Multiple users, through terminals, using a single computer at the same time.                                        |
| _____ | 11. Software, sometimes called the monitor or executive program.                                                        |
| _____ | 12. Computer instructions written in pseudolanguage such as COBOL, FORTRAN, or BASIC.                                   |
| _____ | 13. A waiting line file used for storing either data or programs.                                                       |
| _____ | 14. Input/Output devices used in electronic data processing.                                                            |
| _____ | 15. An addressable portion of a program or data.                                                                        |

**BCPI/Computer Hardware Utilization**  
**Unit 3: Computer Operation and Maintenance**  
**Teacher's Guide/COMPETENCY TEST**

- \_\_\_\_\_ 16. A procedure initiated by a control card or command entered through a terminal or console typewriter.
- \_\_\_\_\_ 17. Direct communication between the terminal user and the computer occurs.
- \_\_\_\_\_ 18. A collection of programs, routines, and, in some cases, source code stored on magnetic disk.
- \_\_\_\_\_ 19. The hardware that coordinates the movement of data between the CPU and an I/O device.
- \_\_\_\_\_ 20. Responsible for performing mathematical operations.

## **PART II: OPERATING AND MAINTAINING MACHINES**

### **Project 1**

**INSTRUCTIONS:** List three causes of execution-time interrupts.

1.

2.

3.

### **Project 2**

**INSTRUCTIONS:** List and describe the start-up procedures involved in operating a computer in your classroom.

### **PART III: MONITORING AND REPORTING ENVIRONMENTAL PROBLEMS**

**INSTRUCTIONS:** Complete the following questions.

1. List three environment variables.
  - a.
  - b.
  - c.
2. Identify three types of fire-prevention controls.
  - a.
  - b.
  - c.
3. Identify four hazard priorities.
  - a.
  - b.
  - c.
  - d.
4. Describe three means of fire prevention.
  - a.
  - b.
  - c.

## **PART IV: INVENTORY RESOURCES AND REPORT FINDINGS**

### **Project 1**

**INSTRUCTIONS:** Define the following inventory terms.

1. Quantity-on-hand
2. Quantity-on-order
3. Reorder point
4. Reorder quantity
5. Physical inventory

### **Project 2**

**INSTRUCTIONS:** List three types of dated materials and describe any special handling or procedures necessary with them.

- 1.
- 2.
- 3.

## PART IV (Continued)

### Project 3

**INSTRUCTIONS:** Identify five areas to be considered when organizing items for storage.

- 1.
- 2.
- 3.
- 4.
- 5.

## PART V: HANDLING MEDIA AND SUPPORT TOOLS

### Project 1

**INSTRUCTIONS:** Describe five diskette-handling procedures.

- 1.
- 2.
- 3.
- 4.
- 5.

### Project 2

**INSTRUCTIONS:** Describe the procedures for formatting a diskette.



**PART V (Continued)**

**Project 3**

**INSTRUCTIONS:** List the routines that should be completed daily in order to handle media correctly.

## PART VI: MONITORING HARDWARE PERFORMANCE

**INSTRUCTIONS:** Identify the meaning and contents of each part of the error messages provided by your instructor. Describe the areas on the screen where the error messages are displayed.

## PART VII: SCHEDULING WORKFLOW

**INSTRUCTIONS:** From the run instructions for a sample job provided by your instructor, determine:

1. Resources needed (paper, magnetic media, etc.)
2. Distribution
3. Special instructions for exception reporting, restart, and programmed messages
4. Data controls

## PART VIII: MAINTAINING DATA AND PHYSICAL SECURITY

**INSTRUCTIONS:** Answer the following questions.

1. List three physical threats to data files.
  - 1)
  - 2)
  - 3)
2. Briefly describe what should be done to protect against loss of data.
3. Briefly discuss how business ethics affect you in a programming setting.

## ACTIVITY ANSWERS

### Activity 1

```

F I      N
APPLICATIONPROGRAM
M L O    D
I  COMPUTEROPERATOR
L O P
Y R IPOSWITCH I S
O E L C      N Y
F I E P    A TASK
C M R      OFFLINE R M
O A J      U R E U
M G L V P   P S L
P DEFACTOSTANDARD T
U L          G E I
T I LINKAGEEDITOR T
E B          E A
R R          OVERLAYS
S A J      S K
FOREGROUND MEMORY I
Y B              N
                POLLING
    
```

**Activity 2** Evaluated to the satisfaction of the instructor.

### Activity 3

**Project 1** Evaluated to the satisfaction of the instructor.

**Project 2** Evaluated to the satisfaction of the instructor.

### Project 3

|    |     |     |     |
|----|-----|-----|-----|
| 1. | 121 | 6.  | 182 |
| 2. | 24  | 7.  | 83  |
| 3. | -5  | 8.  | 58  |
| 4. | -18 | 9.  | 1   |
| 5. | 32  | 10. | -9  |

**Projects 4-6** Evaluated to the satisfaction of the instructor.

#### Activity 4

##### Project 1

- |    |                 |    |                |
|----|-----------------|----|----------------|
| 1. | Stock forms     | 5. | Disposition    |
| 2. | Special forms   | 6. | Identification |
| 3. | Quantity needed | 7. | Open boxes     |
| 4. | Location        |    |                |

##### Project 2

Variations may occur because of pilferage, incorrect entries, etc.

Projects 3-5 Evaluated to the satisfaction of the instructor.

#### Activity 5

Projects 1-6 Evaluated to the satisfaction of the instructor.

Project 7 Answers may vary. Possible answers may include:

1. Date
2. Disk name
3. File name
4. Originator

**Activity 6** Answers may vary according to the messages provided.

#### Activity 7

Projects 1-2 Evaluated to the satisfaction of the instructor.

#### Activity 8

Projects 1-2 Evaluated to the satisfaction of the instructor.

## COMPETENCY TEST ANSWERS

### Part I

- |                     |                           |
|---------------------|---------------------------|
| 1. Computer system  | 11. Supervisor            |
| 2. JCL              | 12. Source code           |
| 3. Microcomputer    | 13. Queue                 |
| 4. Operating system | 14. Peripheral            |
| 5. Register         | 15. Page                  |
| 6. System           | 16. Job                   |
| 7. Task             | 17. Interactive computing |
| 8. Throughput       | 18. Library               |
| 9. Utility          | 19. Controller            |
| 10. Time sharing    | 20. ALU                   |

### Part II

#### Project 1

1. Memory dumps
2. Abnormal ending reports
3. Restart procedures
4. Determine console messages
5. Status of indicator lights

#### Project 2

Evaluated to the satisfaction of the instructor. Answers will vary according to the hardware used.

### Part III

- |     |               |
|-----|---------------|
| 1.1 | Humidity      |
| 1.2 | Temperature   |
| 1.3 | Dust/Dirt     |
| 2.1 | Sprinklers    |
| 2.2 | Sand          |
| 2.3 | Extinguishers |

**Part III (Continued)**

- 3.1 People
- 3.2 Property
- 3.3 Data
- 3.4 Equipment
  
- 4.1 Designated smoking areas
- 4.2 Storage of combustibles
- 4.3 Electrical circuits

**Part IV**

**Project 1**

- 1. The amount of the item that is available.
- 2. The amount of the item that has been ordered.
- 3. The point at which the item needs to be reordered.
- 4. The amount to be reordered to keep an adequate amount on hand.
- 5. Actually taking a hand count of the items rather than relying on inventory.

**Project 2** Descriptions may vary and should be evaluated to the satisfaction of the instructor. Any of the following may be included:

- 1. Printer ribbons
- 2. Felt markers
- 3. Dated forms
- 4. Cleaning chemicals
- 5. Adhesive labels

**Project 3** Descriptions may vary and should be evaluated to the satisfaction of the instructor. Any five of the following may be included:

- 1. Weight
- 2. Bulk
- 3. Frequency of access
- 4. Stability
- 5. Shelf life
- 6. Companion items
- 7. Value



**Part V**

**Project 1** Other answers may be appropriate and should be evaluated to the satisfaction of the instructor.

1. Use the rule of thumb--pick up disk with thumb on the label.
2. No food or drink around area.
3. No magnets near diskettes--phones, radios, etc.
4. Do not bend.
5. Keep in moderate temperature away from extreme temperatures.
6. Do not write on the diskette.

**Project 2** Answers may vary according to the equipment and operating system used.

**Project 3** Answers will vary according to media used, equipment, and instructor's lecture.

**Part VI** Evaluated to the satisfaction of the instructor.

**Part VII** Evaluated to the satisfaction of the instructor.

**Part VIII**

1. Fire, natural disaster, environmental problems, sabotage.
2. Backing up files should be done frequently and regularly. The storage medium should also be checked regularly.
3. Evaluated to the satisfaction of the instructor.

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(2)(A)(D)

Students shall be provided opportunities to review computer hardware operations and review the proper care of hardware and software.

---

| OBJECTIVE: | The students' knowledge will be reinforced<br>in all phases of operating computer<br>hardware. | MASTERY | NON-<br>MASTERY |
|------------|------------------------------------------------------------------------------------------------|---------|-----------------|
|------------|------------------------------------------------------------------------------------------------|---------|-----------------|

---

|    |                                                                     |       |       |
|----|---------------------------------------------------------------------|-------|-------|
| 1. | Identify terminology related to computer operation and maintenance. | _____ | _____ |
|----|---------------------------------------------------------------------|-------|-------|

|    |                                |       |       |
|----|--------------------------------|-------|-------|
| 2. | Operate and maintain machines. | _____ | _____ |
|----|--------------------------------|-------|-------|

|    |                                            |       |       |
|----|--------------------------------------------|-------|-------|
| 3. | Monitor and report environmental problems. | _____ | _____ |
|----|--------------------------------------------|-------|-------|

|    |                                          |       |       |
|----|------------------------------------------|-------|-------|
| 4. | Inventory resources and report findings. | _____ | _____ |
|----|------------------------------------------|-------|-------|

|    |                                 |       |       |
|----|---------------------------------|-------|-------|
| 5. | Handle media and support tools. | _____ | _____ |
|----|---------------------------------|-------|-------|

|    |                               |       |       |
|----|-------------------------------|-------|-------|
| 6. | Monitor hardware performance. | _____ | _____ |
|----|-------------------------------|-------|-------|

|    |                    |       |       |
|----|--------------------|-------|-------|
| 7. | Schedule workflow. | _____ | _____ |
|----|--------------------|-------|-------|

|    |                                      |       |       |
|----|--------------------------------------|-------|-------|
| 8. | Maintain data and physical security. | _____ | _____ |
|    |                                      | _____ | _____ |
|    |                                      | _____ | _____ |

STUDENT NAME \_\_\_\_\_

## SECTION II: COMPUTER HARDWARE UTILIZATION

### UNIT I: COMPUTER HARDWARE

#### UNIT

**OBJECTIVE:** Students will reinforce their knowledge of all phases of operating computer hardware.

#### STUDENT

**COMPETENCIES:** After completing this unit, the student will be able to:

1. Identify the types of computers.
2. Describe primary memory.
3. Describe various input devices.
4. Identify various output devices.
5. Identify other hardware devices.

#### FOCUS:

Being familiar with all phases of computer hardware operation is valuable in most businesses.

## TERMINOLOGY

|                                        |                                                                                                                                                                                                                                                                                                                      |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Arithmetic and logicality (ALU)</b> | The ALU consists of registers and logic circuitry and is responsible for performing mathematical operations.                                                                                                                                                                                                         |
| <b>Buffer</b>                          | An area reserved for data either coming into or leaving main memory. The data are stored in the buffer until the information can be processed or, in the case of the output buffer, until the output device is ready for the data.                                                                                   |
| <b>Central control unit (CCU)</b>      | The CCU is responsible for fetching an instruction, breaking the instruction into parts, moving the necessary data into the ALU, and seeing that the instruction is executed.                                                                                                                                        |
| <b>Central processing unit (CPU)</b>   | The mainframe of the computer contains the CPU, ALU, main memory, input control unit, and output control unit. Sometimes the term CPU is used to denote the computational ability of the computer rather than the five components of the mainframe.                                                                  |
| <b>Channel</b>                         | A physical device that transmits data from an I/O device to the CPU or from the CPU to an I/O device.                                                                                                                                                                                                                |
| <b>Computer operator</b>               | The individual responsible for setting up the computer system to execute jobs, responding to messages generated, and monitoring the computer system in order to achieve maximum throughput.                                                                                                                          |
| <b>Computer system</b>                 | A computer system, often called a system, consists of the CPU, I/O devices that are used to input and output data, and the operating system.                                                                                                                                                                         |
| <b>Controller</b>                      | The hardware that coordinates the movement of data between the CPU and an I/O device.                                                                                                                                                                                                                                |
| <b>Control unit</b>                    | Hardware that includes controlling circuitry for the attached input or output devices (the controller).                                                                                                                                                                                                              |
| <b>Dumb terminal</b>                   | A terminal that cannot be programmed. It is always under the direct control of the host computer. It cannot do any processing on its own.                                                                                                                                                                            |
| <b>Family of computers</b>             | A group of computers that have the same general architecture. At the low end of the family, a limited amount of memory and slow I/O devices are available. At the high end of the family, the I/O devices are much faster, more memory can be obtained, and a wider range of software products is usually available. |

|                              |                                                                                                                                                                                                                                                                                                                         |
|------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Front-end processor</b>   | An interface between the terminals and the computers. The front-end processor, or controller, may only assemble the bits transmitted into characters and provide a parity check. Other front-end processors are mini- or microcomputers and assist the host computer with routine functions required in teleprocessing. |
| <b>Input/output (I/O)</b>    | Computer hardware designed to enter data devices into a computer system and to receive the results of the processing of data.                                                                                                                                                                                           |
| <b>Interactive computing</b> | Direct communication between the terminal user and computer occurs.                                                                                                                                                                                                                                                     |
| <b>Interrupt</b>             | Due to a normal or an abnormal occurrence, control of the system's resources returns to the supervisor.                                                                                                                                                                                                                 |
| <b>Interrupt</b>             | A basis for multiprogramming when the CPU processes a program until it requires an I/O operation.                                                                                                                                                                                                                       |
| <b>IPO switch</b>            | Immediate Power Off switch. A switch used in the case of an emergency to cut off the computer's power.                                                                                                                                                                                                                  |
| <b>Memory management</b>     | The allocation of memory.                                                                                                                                                                                                                                                                                               |
| <b>Microcomputer</b>         | A computer containing a CPU on a single chip. It is a set of Large-Scale Integration (LSI) circuits capable of performing the essential function of a computer. Within a microcomputer, other chips and circuit boards are used to control I/O activities and to add memory to the system.                              |
| <b>Multiplexer channel</b>   | A connection between the control unit and the computer. Data from two or more I/O devices are interleaved a byte, or a block, at a time while being transferred between the control unit and the computer.                                                                                                              |
| <b>Node</b>                  | A computer that provides the interface between terminals and the software supporting communication devices and terminals.                                                                                                                                                                                               |
| <b>Off-line</b>              | A data processing operation or device that is not under the direct control of the computer.                                                                                                                                                                                                                             |
| <b>On-line</b>               | An operation or device under the direct control of the computer.                                                                                                                                                                                                                                                        |
| <b>Operating system</b>      | Software that controls the execution of computer programs. It may provide for scheduling, debugging, I/O control, library management, error routines, and many other related services. It runs the computer as compared to application programs that run on the computer.                                               |

|                             |                                                                                                                                                                                                                                                                                                                                            |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Operations</b>           | The area within the data processing department that involves data entry, computer and peripheral operations, job scheduling, library maintenance and management, and many other related services.                                                                                                                                          |
| <b>Page</b>                 | An addressable portion of a program or data.                                                                                                                                                                                                                                                                                               |
| <b>Page frame</b>           | An area in real memory used to store one page of instruction or data. Usually 2K or 4K.                                                                                                                                                                                                                                                    |
| <b>Peripheral equipment</b> | Input or output devices used in electronic data processing.                                                                                                                                                                                                                                                                                |
| <b>Port</b>                 | An entry channel to which a dataset is attached. One port is assigned to each user of the system.                                                                                                                                                                                                                                          |
| <b>Problem state</b>        | The CPU is executing instructions which are part of an application program.                                                                                                                                                                                                                                                                |
| <b>Register</b>             | A temporary storage device used for special functions such as storing the instructions being executed and the address of the next instruction. Registers are also used for the arithmetic and logic operations.                                                                                                                            |
| <b>Reliability</b>          | The percentage of time that the computer is up relative to the time it is scheduled to be up and running.                                                                                                                                                                                                                                  |
| <b>SCP</b>                  | Systems Control Program is another term used to denote operating system software.                                                                                                                                                                                                                                                          |
| <b>Selector channel</b>     | A connection between the control unit and one type of high-speed I/O device.                                                                                                                                                                                                                                                               |
| <b>Smart terminal</b>       | A terminal that has its own microcomputer which can be programmed. The terminal can function either on- or off-line.                                                                                                                                                                                                                       |
| <b>SPOOL</b>                | Simultaneous Peripheral Operations On-Line. A technique by which slow I/O operations are accomplished faster and more efficiently by first storing the data on a storage device, usually a disk, and then processing the data directly from the intermediate storage area. I/O operations may occur at the same time as program execution. |
| <b>Supervisor</b>           | Software that is sometimes called the monitor or executive program. The supervisor is stored within the CPU and coordinates all other software that make up the operating system.                                                                                                                                                          |
| <b>System</b>               | An organized method for achieving a given result.                                                                                                                                                                                                                                                                                          |
| <b>Unbundled</b>            | A term denoting individual pricing of hardware, software, and support services.                                                                                                                                                                                                                                                            |

## UNIT OVERVIEW

- I. **Types of computers**--computer systems vary widely in how the hardware is designed. However, what the hardware does is similar in all systems. Size is the most basic difference among computer systems. Five computer systems are listed below.
  - A. **Supercomputers**--large computers with fast processing speeds that can access billions of characters of data.
    1. The **CRAY** is one of the best-known examples of supercomputers. The **CRAY** sells for about \$20 million. The services of two full-time engineers are included in the purchase price. Another example of a supercomputer is the **CYBER**.
    2. Both are used in such applications as nuclear physics, meteorology, and petroleum engineering.
    3. Installation requires a specially prepared subflooring to support the weight of the computer and special plumbing to carry the fluorocarbon fluid needed as coolant.
    4. About 150 supercomputers have been installed through 1987.
  - B. **Mainframes**--computers built to minimize the distance between points for very fast operations. Used for extremely complicated computations. Often require a specially designed room with air conditioning and a subfloor which covers the cable that connects the processor with the input and output devices.
    1. Cost over \$400,000 and can exceed \$1,000,000.
    2. Serve more than one user, because they support large networks of individual terminals and remote job-entry locations.
    3. Used by banks, large commercial and industrial companies, and government agencies. Most professional programmers work in a mainframe environment.
    4. The typical mainframe configuration is the IBM System/370. The IBM System/370 is a family of upward-compatible, general purpose computers used for a variety of data processing applications that provide a wide range of performance levels and processor storage capacities. The individual computers of this family are referred to as models or processor complexes.
      - a. The **central processing unit (CPU)** is the controlling center of the computer. The CPU's job consists of:
        - 1) Addressing main storage.



- 2) Retrieving and storing data.
  - 3) Arithmetic and logical processing of data.
  - 4) Executing instructions in a desired sequence.
  - 5) Initiating communication between main storage and input/output devices.
- b. **Main storage** provides the system with direct, addressable, and fast access to stored data.
- c. **Operator facilities** provide for manual operation and control of the system. They include:
- 1) Operator-to-machine communication.
  - 2) Indication of machine status.
  - 3) Control over the setting of the time-of-day clock.
  - 4) Initial program loading.
  - 5) Resets and other manual controls for operator intervention in normal machine operation.
  - 6) This is all done through the system console, a dedicated screen and keyboard.
- d. **Channels** are the direct controllers of Input/Output devices and control units.
- e. **Input/output devices** fall into a number of categories, some of which overlap. They are used in and for:
- 1) **Auxiliary storage**--machine and manual input.
  - 2) **Data communications**--reading of external documents and displays.
  - 3) Process control.
  - 4) Data acquisition.
- f. **Input/output interface** provides a uniform method of attaching various input/output (I/O) devices (through control units) to a channel. The information format and the control signal sequences provided by the interface are independent of the type of control unit and channel. Input/output operations transfer data between main storage and I/O devices.



- g. **Control unit function** provides the logic circuitry and the storage areas (buffers) needed to operate the attached I/O devices.
- C. **Minicomputers**--moderate-sized computers which occupy their own floor space. Used when a desktop computer is not powerful enough to do the job.
  - 1. Can be relocated conveniently, because they have no special power or environmental control requirements.
  - 2. Smaller, slower, and less expensive than mainframes but perform the same tasks as mainframes (only on a reduced scaled).
  - 3. Cost between \$10,000 and \$40,000.
  - 4. Store and retrieve data from the same type of input and output devices as mainframes do, but with a smaller storage capacity.
  - 5. A small business system is often part of an upward-compatible family of computers. IBM System/36 is an example of a small business system and may have the following features:
    - a. **Hardware**--the CPU consists of two computing elements, the system processor, which performs all system-level functions, and the execution processor, which performs all computations.
    - b. **Storage**--the memory is 640K and up. Fixed-disk drives; two fixed each with the ability to store 27.1 megabytes. The diskette drive can hold up to three individual diskettes and up to two magazines. Each magazine holds up to ten diskettes.
    - c. **Peripherals**--the printer 5211 Model 2 has 48 different characters; approximately 300 lines per minute can be printed. Also workstations can be attached. The operator communicates with the computer by responding to system- or program-generated messages displayed on the CRT. Data can also be entered from the operator's console.
    - d. The system can be expanded by adding more memory, additional fixed-disk drives, disk drives with higher storage capacity, more workstations with faster printers, additional diskette drives, a communication adapter, and workstations to be added at remote locations.
    - e. **System support product (SSP)** provides the function of the operating system.
    - f. **Utilities programs product (UPP)** assists in maintaining files, entering data, formatting screens and sorting files.

- g. The programmer writes all in-house programs in RPG. COBOL, BASIC, and assembler language are also available.
  - h. Interactive communication feature permits the system to work in an interactive distributed network.
- D. **Microcomputers**--also known as personal computers or desktop computers; they are largely owned by individuals. The name derives from its main computing component which is located on one integrated circuit of chip.
- 1. Limited in the volume of data they can process.
  - 2. Fit on a desk or table.
  - 3. Do not require a data processing expert to operate them.
  - 4. Support peripheral devices.
  - 5. Types of microcomputer systems:
    - a. IBM Compatible
      - 1) The 8088 (XT) is a 16-bit processor with an 8-bit data bus. It is capable of addressing 640K of memory and is the oldest and slowest of the IBM compatibles, but it is effective for many applications. It usually runs at 4.77 to 10 MHz and is the most common processor for business applications.
      - 2) The 8086 is a 16-bit processor with a 16-bit data bus. It usually runs at 8 to 12 MHz and is a faster processor. The PS/2 models 25 and 35 fall into this category.
      - 3) The 80286 (AT) is the business standard for effective data processing on a microcomputer. It is a 16-bit processor running at 6-20 MHz.
      - 4) The 80386 is the fastest and best microprocessor available. It is a 32-bit processor running at 16-25 MHz with a 32-bit data bus. It is very fast and efficient, even for complicated applications like graphics and CAD.
    - b. The Apple family of microcomputers
      - 1) The Apple IIe is a well-known and very functional microcomputer in the area of education. Business applications are, however, somewhat limited. The Apple IIe uses a 16-bit processor, the Motorola 6502, with an 8-bit data bus.
      - 2) The Apple IIgs is a larger and faster computer but still not widely used in business. It also uses the Motorola

6502 processor but adds additional memory and a color monitor.

- 3) The Macintosh is a very fast and functional microcomputer. It uses the Motorola 6800 series processor, with a 32-bit chip. It comes with a high resolution color monitor, a mouse, and has a graphics interface, which makes it very easy to use. Though not as widespread as the IBM compatible, the Macintosh is often used for business applications.

E. **Portable computers**--or laptops as the smaller portable ones are called. These computers have all the features of the larger computers.

1. Some have hard drives and others use the 3.5 floppy disk.
2. *Footprint* is the term to refer to the amount of desk space the computer will require. The average footprint of a portable is 11 inches by 12 inches, but this can vary. They weigh between 5 and 20 pounds.
3. The screens use a liquid crystal display (LCD) similar to most digital watches.
4. Some portables have a built-in power supply that is usually rechargeable.
5. The same programs and files can be shared by a portable and a microcomputer; this is assuming the machines are compatible.

## II. Hardware components

A. **Central processing unit (CPU)** can be thought of as the brain of the computer. Traditionally, it is one processor doing one job at a time in one computer.

1. The common processors
  - a. **MS-DOS machines**--Intel 8088, 8086, and 80286--16-bit; and the Intel 80386--32-bit.
  - b. Apple family of computers--Macintosh, Motorola 6800 series--32-bit; Apple IIe, Motorola 6502--16-bit.
2. Parts of the CPU
  - a. The **clock** is a crystal that vibrates at a high frequency. Clock speed is measured in millions of cycles per second, or megahertz (MHz). Common clock speeds vary from 1.7 to 24 MHz.

- b. The bus is the nervous system of the computer. The bus connects the CPU to itself and to the other components of the system. The bus connects many different points at once, but communication can only take place one message at a time. This is why bus width is important. The width of the bus is the number of bits that it can carry at once. The following are some common widths and the computers in which they are used: 8-bit--XTs and Apple IIs; 16-bit--ATs and lower PS/2 line; and 32-bit--386's and upper PS/2 line.
- c. Registers are memory locations within the CPU where the CPU holds data for processing. The larger the registers, the larger and more powerful the CPU. The common sizes are 8-, 16-, and 32-bit. Each CPU has a number of registers, some for special purposes:
  - 1) The instruction register holds the location of the next instruction to be executed.
  - 2) The decoding unit decodes instructions read from memory.
  - 3) The accumulator is the main register for containing data which are being manipulated.
  - 4) The working register and additional registers hold the data and addresses being manipulated.
- d. **Arithmetic/logic unit (ALU)** is used for some instructions required for arithmetical or logical operations to be performed on data.

**B. Primary memory**

- 1. There are several functions of primary memory.
  - a. Stores the program being executed.
  - b. Stores the data being used by the program.
  - c. Stores the operating system of the computer.
  - d. Serves as a buffer between fast and slow operations.
  - e. Represents screen images.
- 2. Memory is organized into sequence of locations.
  - a. The locations are numbered--each number is called an address.
  - b. The byte or word in an address is called that address's content.

3. Types of primary memory
  - a. **Read-only memory (ROM)** cannot be changed. It is set at the factory by the manufacturer. ROM includes such operations as how to start the computer and how to handle the screen.
  - b. **Random-access memory (RAM)** can be changed. The processor can read and write this memory. RAM must have continuous power to keep its data intact. Because of this, it is referred to as volatile. RAM is stored in chips. These chips are grouped in multiples of eight. Common sizes for microcomputers are 64K and 256K.
4. Memory is measured in K, or kilobytes. Secondary memory is recorded memory that the processor can access to perform its programming. Some common sizes for microcomputers are:
  - a. 64K is considered adequate for basic applications.
  - b. 128K is much more versatile.
  - c. 312K is the lower end for professional quality word processors and business programs.
  - d. 512K is a good figure for average word processing and business programs.
  - e. 640K is the limit of MS-DOS and the standard for most higher-level graphics and desktop publishing programs.
  - f. 1MB (1000K) is standard for most AT-based programs and the IBM PS/2 line. It is the low-end mark for effective multitasking (running more than one program at once).
  - g. 2MB and up are required for some graphics, multitasking, windows, PS2, and other advanced programs
- C. **Secondary memory**--this is recorded memory that the processor can access to perform its programming. **On-line** storage is memory that can be reached electronically; **off-line** is memory that requires mechanical assistance.
  1. Hard disk
    - a. Hard disks were developed by IBM. The early ones were called Winchester disk drives, because their storage capacity, two disks at 30 MB each, was similar to the caliber of the famous Winchester rifle (30/30). Standard sizes of 20 to 370 MB.

- b. Access time is the time it takes to retrieve a specific piece of information and is usually measured in special clean environments which must remain sealed throughout their use to reduce the possibility of contamination.
  - c. The read/write head flies over the surface of the hard disk at about  $1/1,000,000$  of an inch. A particle of dust would be 100 times this size. Hard disks must be manufactured in special clean environments and must remain sealed throughout their use to reduce the possibility of contamination.
  - d. Whenever a hard disk is moved, the DOS command "Park" should be issued so that the read/write head can be positioned in a safe place to avoid crashing into the surface of the disk and possibly damaging the disk or the information on the disk.
2. Floppy disk
- a. Made of mylar plastic which is coated with a surface that can record magnetic images.
  - b. Housed inside a protective jacket.
  - c. Inexpensive and quite durable.
  - d. The 5.25-inch floppy has a central hole called the hub. The disk drive grips the disk so that it can be spun during the reading and writing process.
  - e. The 5.25-inch floppy has a notch on the edge near a top corner. This is the write-protect notch. This notch, if open, allows you to write data on the disk and to erase data from the disk. To prevent unwanted writing to or erasing of files, an adhesive tab can be placed over the notch.
3. Disk storage structure
- a. Information is stored in concentric rings called tracks.
  - b. Each track is organized into a number of sectors, each separated by a space. Some disks come with sectors prerecorded at the factory. This is called a "hard-sectored" disk. Disks without prerecorded sectors are called "soft-sectored" disks. Soft-sectored disks must be formatted before they can be used.
4. There are several sizes of disks available.
- a. Currently the 5.25-inch floppy disk is the most popular for the microcomputer. It is used on both the Apple and MS-DOS



machines. The capacity for an Apple disk is 140K, and the capacity for a MS-DOS disk is 360K or 1.2 MB.

- b. The 3.5-inch floppy disk is quickly taking over the market place. It has a plastic case and is sometimes called a "stiffy." It has a capacity of 720K or 1.44MB.
- c. For the mainframes, the types of disks can be removable, non-removable, fixed-head drives, or data modules.

5. File organization methods

- a. Sequential
- b. Random or direct
- c. Indexed

D. **Monitors/cathode ray tube (CRT)/video display terminal (VDT)**--are the most widely used output devices for personal computers. For a microcomputer, one terminal is most likely; however, for larger computers, hundreds of terminals can be supported.

1. Features to consider are:

- a. The screen resolution is an important consideration. *Resolution* means how many dots are used to make up an image. The computer screen is composed of many little dots called "pixels." The pixels visually blend together to form an image. The more pixels there are on a screen, the clearer that screen's images will be.
- b. Color monitors must use three pixels. These three pixels are three different colors: red, green, and blue. The composite color image is formed by turning on these three primary colors in various combinations and intensities. Using three pixels in place of one decreases the resolution. To remain at a comparable level of resolution with the monochrome monitor, the color monitor must have three times as many pixels.

2. Types of monitors available

- a. **IBM Color Graphics Adapter (CGA) monitors** have the lowest resolution. CGA monitors will display four colors at a resolution of 640 by 200 pixels. This is adequate for some applications. The CGA monitors are the least expensive color monitors.

- b. **EGA monitors** are good quality color monitors. EGA stands for Enhanced Graphics Adapter. It is the next step up in display resolution. The EGA display will show any 16 colors at once out of a total of 64 colors, at a resolution of 640 by 350 pixels.
  - c. **Hercules monochrome monitors** have similar resolution to the EGA. This monitor is driven by the Hercules compatible graphic adapter. It provides a resolution of 722 by 350 lines in two colors. This display has acceptable resolution for most applications and is by far the most cost-effective display system available, with many companies selling both monitor and display adapters as a kit for under \$100. Monitors are available in either amber on blue, green on white, or black on white.
  - d. **Vector graphic array (VGA) monitors** have the highest resolution currently available. These monitors start at a resolution of 640 by 480 pixels and can go up to around 1200 by 1000 pixels. VGA monitors can display 256 colors out of a palette of at least 4000. A new display mode called Super VGA has a resolution of 800 by 600 pixels.
- E. **Input devices**--are a number of ways to get information into the computer.
- 1. **Keyboards** are the classic input device for computers.
    - a. Standard XT keyboards have 88 keys. The letters and numbers are laid out like a typewriter. The function keys are on the left, and the arrow keys are on the right.
    - b. Enhanced 101 key or AT keyboards have several extra keys, including an extra set of number keys arranged on a number pad so that entering data through the ten-key by touch method is simplified. The AT keyboard also adds two extra function keys and places all function keys in a single row above the regular number keys.
    - c. Apple keyboards do not have function keys; however, they do have the "open Apple" and "closed Apple" keys, which in combination with other keys operate as function keys.
    - d. Other keyboard devices capture key information onto magnetic tape or magnetic disk.
  - 2. **Magnetic ink character recognition(MICR)**--the system used on all checks and deposit slips. The number at the bottom of the check and deposit slips contain magnetic particles. These can be read by the reader attached to the computers.



3. **Optical character recognition(OCR)**--machines scan documents imprinted with OCR type and data can be read directly into the computer. Mark sensing is a form of OCR.
4. **Universal product code(UPC)**--the bar code on labels can be read by devices that are built into the checkstands. Point-of-sales terminals and wand readers are used to read and record the code of the product purchased.
5. **Magnetic input** is becoming a new method of on-line entry of data into the computers. Information about a customer is recorded on a strip of magnetic tape attached to a plastic card. The magnetic strip can be read into devices that transmit data to computers. The automatic teller machine is an example of this principle.
6. **Tablets**--all lines and points drawn on a tablet are translated into values for the computer.
7. **Mice** are pointing devices used with the computer screen. The mouse is moved along a desktop, and the movements of the mouse on the desktop are digitally matched to the movements of a pointer or cursor on the computer. A screen mouse allows commands to be chosen from on-screen menus, forgoing the necessity of typing in the commands every time they are used.
  - a. **Mechanical mice** track movement by measuring the movements of a rolling ball housed in the puck which is dragged across the surface of a desktop. The mechanical mouse will work on any flat desktop surface.
  - b. **Optical mice** track movement by measuring changes in the puck's position in an underlying coordinate grid in the mousepad. An optical mouse will only work when moved on top of the optical gridpad.
8. **Touch screen** are screens that are touch-sensitive. Three types of touch screens:
  - a. **Pressure-sensitive**--the screens have a matrix of wires that are built-in on mylar sheets. When the surface of the screen is pressed, two wires make contact. This then tells the computer the location that was touched on the screen.
  - b. **Infrared light**--invisible light beams crisscross the surface of the screen in a grid pattern. When the screen is touched, the finger breaks the beam of light. This then signals the computer to respond accordingly.
  - c. **Capacitance**--the screen can sense the change in electrical capacitance when it is touch by the finger.

9. **Light pens** are another way to touch the screen and make something happen. Light pens have been used successfully with the mainframe computer; however, they have never been popular devices for personal computers.
  10. **Joy sticks** are very popular devices for input with the personal computer. These are used largely where games are concerned.
  11. **Voice recognition** screens are slowly making a step forward. They are becoming more reliable. Most voice recognition systems require the user to train the system.
- F. **Printers**--fall in two categories, impact or nonimpact. An impact printer will put the image onto the paper by some type of printing mechanism striking the paper, ribbon, and character together. Nonimpact printers place the image on paper without striking the paper. Some printer types are as follows:
1. **Dot matrix** printers represent characters by selecting certain dots out of an established matrix. This matrix is nine dots high and 12 dots wide. For a higher-quality print, the pattern of dots is moved slightly and repeated, which results in a denser pattern of dots. These darker letters are called near-letter quality.
  2. A **24-pin dot matrix** printer uses two rows of 12 pins to create a matrix that is 12 dots high and 30 dots wide. However, the size of the character is the same size as the characters normally produced by nine-pin printers. The quality approaches that of laser printer, so 24-pin printers are often referred to as letter-quality. Color option kits are also available for 24-pin printers, making them into excellent candidates for graphics applications.
  3. **Daisy wheel** printers use a wheel with letters and characters on the end of spokes which are rotated and hit with a hammer to strike the paper much like a normal typewriter. Daisy wheel printers are an economical source of letter-quality documents but have no graphics capability.
  4. **Laser** printers use the same thermal transfer engine as personal photocopiers. Because of this, their quality is very high. The resolution of the laser printer is 300 dots per inch (DPI). This means that the laser printer constructs text and graphics from a matrix of dots so dense, there would be 9000 (300 by 300) of them in a square inch. Some of the newer ones are in the 600 dots per inch range. This resolution is referred to as "typeset quality."
  5. **Postscript laser** printers can reproduce anything that can be produced on the computer. The Apple Laser Writer was the first postscript printer. It has a page description language that is available on many of today's laser printers.

6. **Ink jet** printers offer similar resolution to that of the laser printer and are less expensive. They are also lightweight and make good portable printers.
  7. **Chain and band** printers are used on mainframe computers. These printers are line printers because a line, which can be up to 132 characters wide, is printed each time. The speed varies from 600 to 3,000 lines per minute.
  8. **Page** printers use electrophotographic and laser technology to print a full page of information. One Honeywell page printing system prints at 600 pages per minute. The cost of a page printer may be up to \$300,000.
- G. **Plotters** use a computer-controlled pen to draw images and are used exclusively for graphics. Pens can be changed to produce almost any color. A plotter can draw any image the computer can produce.
1. **Drum plotters**--with drum plotters, the paper is moved on a rotating drum under the pen to control travel. The pen travels only horizontally.
  2. **Flatbed plotters**--a flatbed plotter holds the paper flat on a horizontal surface, and the plotting pen moves in both directions.
- H. **Modem**--a piece of hardware that can send and receive digital signals over telephone lines. It can be either internal, located within the computer cases, or external, in a separate case of its own. The Hayes compatible is the most widely used type. Modems generally come in three speeds: 300, 1200, and 2400 baud.
1. The two types of modems
    - a. Direct connect
    - b. Acoustic coupler
  2. Modem features may include auto-answer, auto-disconnect, auto-dial, auto-redial, or time-delay.
  3. Methods of transmission may include phone lines, coaxial cables, microwave, satellite or fiber optics.
- I. **Tape drives**--are devices that handle magnetic tape. On the mainframe, a tape drive has two tape reels. One reel is known as the data reel, and the other reel is known as the take-up reel, or the machine reel.
1. Tape files are sequentially organized.
  2. The common size of computer tape reel is 2,400 feet.

3. A computer tape is a ribbon of plastic coated with iron particles that can be magnetized by a recording mechanism built into the tape drive. Data can be recorded on the tape ranging from 400 to 6,250 bytes per inch(BPI). The tape has recording positions which run the entire length of the tape called "tracks," or "channels." The pattern for recording information on the tape is EBCDIC (Extended Binary-Coded Decimal Interchange Code).
4. Tape files are organized into blocks. A block can be one record or many records. Blocks are recorded on the tape by leaving blank spaces between groups of records. These blank spaces are called "gaps."
5. Tape drives are used as part of large computer systems. Microcomputers can use tape cassette drives.
6. Tape units on personal computers are used mainly for backup of the hard disk data. Many times a tape cartridge or a cassette tape is used.

J. Internal hardware

1. **Math coprocessor**--a math coprocessor is a dedicated microprocessor that is especially designed to handle floating point calculations. This frees up the computer's regular microprocessor so computing speed is enhanced. A math coprocessor is essential for CAD and other advanced graphics. Some large spreadsheets may also require math coprocessors.
2. **Extra memory cards**--extra memory cards can be added to enhance the computer's performance. This is often necessary for graphics applications.
  - a. **RAM disk**--one way to use extra memory is through the allocation of part of the extra memory as a RAM disk. A RAM disk can be any size from 64K to 360K. It is used to speed access to the hard disk. Often programs will access the same information many times during the course of running. Accessing a RAM disk is about 100 times faster than accessing a hard disk.
  - b. **Memory cache**--a memory cache works much the same way except that it is smaller, usually 64K, and the program works with it automatically. A memory cache can speed up a system dramatically.

3. **Replacement motherboards**--as the 80286 and 80386 systems became available, so did many replacement motherboards for the old 8088 processor.
  - a. These boards used the same chassis and peripherals as the standard 8088XT system did, except that the 8088 motherboard is replaced with a motherboard which carries the 80286 or 80386 processor.
  - b. This results in a dramatic increase in computing speed and power without the purchase of an entirely new system; however, the computer still has to access input/output devices and peripherals through the XT's old 8-bit data path.
4. **Graphics coprocessor**--a graphics coprocessor is a dedicated processor which frees the central processor from the task of computing information for the display. For high-level CAD and graphics applications, they can result in a dramatic increase in speed and performance.

K. Optical disk

1. Storage systems use the same laser disk technology used in the recording of music on compact digital disk audio systems. The storage capacity is 300MB.
2. Many read-only disks are now available which contain such useful information as the *United States Zip Code Directory*, *Compton's Unabridged Dictionary*, and the *Encyclopedia Britannica*.

L. Bernoulli boxes

1. Devices that are located externally to the computer.
2. Store data on either 10 or 20 MB removable cartridges, allowing unlimited storage.

M. Hard cards

1. Devices which allow 20 to 40 MB of storage to be placed on an expansion port and used as an extra disk drive.
2. Low cost and does not require an open disk drive slot.

## ACTIVITY 1: CLASSIFICATION OF COMPUTER SYSTEMS

### Project 1

**INSTRUCTIONS:** Prepare a notebook of various computer hardware components for: a personal computer, a small-sized business computer, and a mainframe. Include information about CPU, storage, software, peripherals, and memory expansion.

### Project 2

**INSTRUCTIONS:** Research various personal computer configurations. Given the budget provided by your teacher, "purchase" a personal computer. Find advertisements that illustrate the type of systems that you are purchasing. Indicate the price and the characteristics of each component.

### Project 3

**INSTRUCTIONS:** Research various small-sized personal business computer configurations. Working in a group and using the budget provided by your teacher, "purchase" a small-sized business computer system.

### Project 4

**INSTRUCTIONS:** Research various mainframe computer configurations. Bring articles and advertisements to class in order to compile resources to select an appropriate mainframe.

## ACTIVITY 2: COMPONENTS OF COMPUTER SYSTEMS

### Project 1

**INSTRUCTIONS:** List and discuss the major components of the CPU.



## ACTIVITY 2 (Continued)

### Project 2

**INSTRUCTIONS:** Describe the differences between hard disks and floppy disks.



## ACTIVITY 2 (Continued)

### Project 3

**INSTRUCTIONS:** Compare the three types of color monitors.

1.

2.

3.

## ACTIVITY 2 (Continued)

### Project 4

**INSTRUCTIONS:** List and describe five peripheral devices.

1.

2.

3.

4.

5.

## ACTIVITY 2 (Continued)

### Project 5

**INSTRUCTIONS:** Discuss the characteristics of three types of printers.

1.

2.

3.

## SECTION II: COMPUTER HARDWARE UTILIZATION

### UNIT II: COMPUTER SOFTWARE

#### UNIT

#### OBJECTIVE:

Students will understand the role software plays in utilizing computer hardware.

#### STUDENT

**COMPETENCIES:** Upon completion of this unit, the student will be able to:

1. Describe the characteristics and functions of an operating system.
2. Identify and categorize applications that would utilize batch processing and real-time processing.
3. Identify a compiler, interpreter utilities, and virtual storage.
4. Discuss the different versions of the microcomputer operating systems.
5. Identify the major components of the DOS/VS operating system.

#### FOCUS:

A working knowledge of operating systems allows one to be more flexible with the type of computer he/she is able to operate, and this knowledge facilitates the worker in accomplishing basic computer tasks.

## TERMINOLOGY

|                            |                                                                                                                                                                                                                                                                                         |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Application program</b> | Software used to execute the procedures within a system. Programs that run on the computer to perform a particular application.                                                                                                                                                         |
| <b>Batch processing</b>    | The serial execution of programs. Each program is entered into the CPU and is completely executed before a second program is executed.                                                                                                                                                  |
| <b>Book</b>                | The term denotes the source statements, or data, cataloged under a given name.                                                                                                                                                                                                          |
| <b>Boot area</b>           | The MS-DOS disk, also called "boot record," consists primarily of a short machine-language program that starts the process of loading DOS into memory. If it does not find the system files on the disk, it displays the message "Non System Disk."                                     |
| <b>Compiler</b>            | A program used to translate high-level language statements written in a programming language such as COBOL into machine-language instructions.                                                                                                                                          |
| <b>Core dump</b>           | A printout of the content of the computer's memory. Core dumps are sometimes used in debugging programs.                                                                                                                                                                                |
| <b>Core-image library</b>  | A portion of the SYSRES pack reserved for storing ready-to-execute programs such as compilers and application software.                                                                                                                                                                 |
| <b>Defacto standard</b>    | The standard that vendors informally accept, generally because it has come to dominate a market segment of the business.                                                                                                                                                                |
| <b>Event-driven</b>        | A basis for multiprogramming when the CPU processes a program until it requires an I/O operation.                                                                                                                                                                                       |
| <b>FAT area</b>            | File Allocation Table keeps a record of where files are stored and how space is used on the MS-DOS disk.                                                                                                                                                                                |
| <b>Foreground</b>          | An area within the CPU's memory. Some systems' real memory is divided into areas referenced as F1, F2, F3, F4, and BG (background). When a message is displayed pertaining to the execution of a job, the system usually indicates the area (or partition) in which the job is running. |
| <b>High-level language</b> | A language, such as COBOL, which permits the programmer to write source code in English-like statements. The compiler translates each source code instruction to one or more machine-language instructions.                                                                             |

|                                   |                                                                                                                                                                                                                                                  |
|-----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Input/output(IOCS)</b>         | A series of commands for initiating, control system executing, monitoring, and controlling data transfer into and out of the computer.                                                                                                           |
| <b>Interpreter</b>                | A program which checks each source code statement for errors and immediately translates the statement to object code and executes it.                                                                                                            |
| <b>IPL</b>                        | Initial Program Load. Indicators and internal switches are reset, and the supervisor is loaded into the CPU. When a system is IPLed, the operator must depress certain keys on the console and then enter the current date and time.             |
| <b>Job</b>                        | A procedure initiated by a control card or command entered through a terminal or console typewriter. A job can consist of one or several tasks. The end of a job is usually indicated by the use of a JCL card such as the /& card.              |
| <b>Job control language (JCL)</b> | A specialized language for setting up jobs and communicating basic instructions to the computer.                                                                                                                                                 |
| <b>Job entry control language</b> | Control statements used under a software feature such as POWER to provide additional options for the queues.                                                                                                                                     |
| <b>Library</b>                    | A collection of programs, routines, and in some cases, source code stored on magnetic disk.                                                                                                                                                      |
| <b>Linkage editor</b>             | A program supplied by the vendor as part of the operating system that combines the user's object modules with modules from the core-image library. The linking of the user's modules with those from the library produces an executable program. |
| <b>Machine language</b>           | The absolute, or actual, instructions executed by the computer in reading, processing, and writing data.                                                                                                                                         |
| <b>Memory</b>                     | A type of storage used to store programs or data that exist on an auxiliary device.                                                                                                                                                              |
| <b>Multiprocessing</b>            | Two or more CPUs are combined into a single system so that more than one instruction can be executed at one time.                                                                                                                                |
| <b>Multiprogramming</b>           | Two or more computer programs are executed concurrently by a single computer.                                                                                                                                                                    |
| <b>Multitasking</b>               | Operating systems that allow concurrent tasks.                                                                                                                                                                                                   |
| <b>Object code</b>                | Machine-language instructions. After a program has been written, compiled, and debugged, the object code version of the program is often stored in an on-line library.                                                                           |

|                            |                                                                                                                                                                                                                                                                                                                                   |
|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Overlays</b>            | During execution of a program, one section of real memory is reserved for storing program segments. When a section of instructions is executed, a new section is retrieved from the library and overlaid over the section just executed. Overlaying permits a program that requires more memory than is available to be executed. |
| <b>Page-in</b>             | A page of instructions or data is brought into real memory from its location in virtual memory.                                                                                                                                                                                                                                   |
| <b>Page-out</b>            | Restoring a page of data or instructions in real memory. The page frame of real memory is then released and made available for paging in a new page.                                                                                                                                                                              |
| <b>Polling</b>             | A method for inquiring whether a terminal has data to be transmitted.                                                                                                                                                                                                                                                             |
| <b>POWER</b>               | (Priority Output Writer Execution Reader) A software package added to some operating systems that provides for spooling and multiprogramming.                                                                                                                                                                                     |
| <b>Priority system</b>     | The method used to determine which program waiting for execution shall be run.                                                                                                                                                                                                                                                    |
| <b>PSW</b>                 | Program Status Word. The PSW contains, along with other information, the operational code of the instruction being executed and the address of the next instruction.                                                                                                                                                              |
| <b>Query language</b>      | A programming language that permits the user to communicate with the computer in English.                                                                                                                                                                                                                                         |
| <b>Queue</b>               | A waiting line file used for storing either data or programs.                                                                                                                                                                                                                                                                     |
| <b>Real memory</b>         | Storage for data and programs that exists within the central processor's main memory.                                                                                                                                                                                                                                             |
| <b>Real-time</b>           | Data entered into the system is processed in time to affect the activity at hand. Multiple users can simultaneously access the system by means of terminals.                                                                                                                                                                      |
| <b>Relocatable library</b> | A library stored on SYSRES which contains object-code subroutines that can be incorporated into other programs.                                                                                                                                                                                                                   |
| <b>Relocatable phase</b>   | At the time of execution, the program module (or phase) can be loaded into any memory partition.                                                                                                                                                                                                                                  |
| <b>Response time</b>       | The time from when the user issues an inquiry until the computer responds with the answer.                                                                                                                                                                                                                                        |



|                            |                                                                                                                                                                                                                     |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Root directory area</b> | On the MS-DOS disk, the root directory contains basic information about the files stored in the root directory.                                                                                                     |
| <b>Setup deck</b>          | Job control language cards used to initiate a job and to provide required data to execute a job.                                                                                                                    |
| <b>Source code</b>         | Computer instructions written in a pseudolanguage such as COBOL, FORTRAN, or BASIC.                                                                                                                                 |
| <b>Source library</b>      | A library containing statements written in a programming language such as COBOL.                                                                                                                                    |
| <b>SYSGEN</b>              | Generating an operating system which meets the specific needs of the user. From the options available, the user determines which ones are to be included as part of the system's software.                          |
| <b>SYSRES</b>              | The disk pack that must be on-line for the computer system to be operational. Stored in libraries are the user's programs, compilers, error routines, IOCS routines, linkage editor, librarian, and other software. |
| <b>System commands</b>     | Instructions used by operators to communicate with a Times-sharing system. System commands perform the same functions as job control language statements.                                                           |
| <b>Task</b>                | An individual program, or routine, incorporated into a job.                                                                                                                                                         |
| <b>Terminal</b>            | A device for communicating with the computer.                                                                                                                                                                       |
| <b>Throughput</b>          | The amount of data that, within a given period, can be entered into the system, processed, and outputted in final form.                                                                                             |
| <b>Thrashing</b>           | An excessive amount of paging occurs during the execution of a program.                                                                                                                                             |
| <b>Time-sharing</b>        | Multiple users, through terminals, use a single computer at what appears to be the same time. The response time should be such that each user feels that he or she has exclusive use of the system.                 |
| <b>Time-driven</b>         | The basis for multiprogramming where the CPU processes programs on a time-sharing basis, giving each program a slice of time.                                                                                       |
| <b>Transient area</b>      | An area set aside in the memory of the computer for error routines, messages to the operator, and other types of control software.                                                                                  |
| <b>Utility programs</b>    | Frequently used programs that perform tasks such as copying files, sorting files, providing backups for tapes and disks and providing tape-to-disk conversions.                                                     |



|                 |                                                                                                                                                                                                                                                                                 |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Virtual memory  | Storage for programs or data that exist on an auxiliary device such as a magnetic disk. Virtual memory is considered an extension of real or main memory. During execution of a program, data or parts of a program are swapped back and forth between real and virtual memory. |
| Virtual storage | Real memory is expanded to include system virtual memory. In a VS system, only the portion of the program executed needs to reside in real memory.                                                                                                                              |
| VS              | Addressable spaces on a secondary storage device (usually some type of fast-swapping memory) that appears to the programmer as real memory. From VS memory, instructions and data are swapped into real memory for execution.                                                   |
| VSE             | Virtual Storage Extended. Operating system software with features normally not included in a typical VS system.                                                                                                                                                                 |

## UNIT OVERVIEW

- I. **System software**--is a series of programs that have been written to simplify the interface between the programmer and the computer. These programs run the computer, as compared to application software, that runs on the computer. System software is generally composed of operating systems, language translators, utilities, and user interface shells.
  - A. **Characteristics of operating systems**--during the era of second-generation computers, their arithmetic and logic ability was not challenged. Often a computer's computational ability was used only 10% of the time. Management was concerned about a powerful and very expensive computer seemingly sitting idle 90% of the time. More throughput had to be achieved. The computer should not need to wait for the operator to set up jobs. IOCSs (Input/Output Control System) developed during the second generation made programming easier.
    1. IOCS (Input/Output Control System) might be considered the pioneer operating system.
      - a. Today the input/output control commands are stored in an on-line library.
      - b. I/O control initiates, executes, monitors, and controls the transfer of data into and out of the computer.
      - c. I/O control eliminates details of handling I/O operations. Since separate I/O routines need not be written by each programmer, uniformity is achieved.
    2. Basic functions of an operating system
      - a. **Job management** is responsible for managing the total computer system. Communication between computer and operator is provided as part of the job-management function.
      - b. **System resource management** allows the resources of the system to be allocated to programs being executed. These resources include the computer's memory, computational ability, and I/O devices.
      - c. **Data management** is creating, changing, and reorganizing files. Routines within the operating system must provide for utilization of space on the direct-access devices, allocation of the tape drives, reading of cards, and output of printed material. Usually the operating system controls all I/O operations.

3. Overall function of the operating system is to control the activities of the computer system. It serves as the traffic cop, directing and managing computer events.
  - a. The **supervisor** is a set of programs to handle the overall management of the many jobs and tasks that are being conducted by the computer system.
  - b. **Resource management** is a function concerned with making the computer's resources such as the CPU, primary memory, input/output devices, and support software available in an efficient way.
  - c. **Data communication** is the transfer of data between primary memory and peripheral devices.
  - d. **Command language** is a series of commands that allows you to use the computer.
4. Types of operating systems
  - a. Batch processing jobs are usually entered from the card reader or terminal and executed one job at a time in the sequence entered.
    - (1) **Operating system** is used for scheduling I/O and for handling problems that occur during the execution of the jobs.
    - (2) Many small computer systems start with an operating system that handles only serial batch jobs. Usually, numerous off-line operations are used to prepare the data for processing. As the needs of the business increase, a more complex operating system is obtained.
  - b. Multiprogramming allows the computer to run two or more jobs at the same time.
    - (1) Productivity of the computer is increased, because the CPU is used more efficiently. For example, while one job is waiting for an I/O event, another job is using the CPU to execute its instruction.
    - (2) Under multiprogramming, only one instruction at a time can be executed. This is called "concurrent processing," not simultaneous.

- (3) Multiprogramming compensates for the difference between the high processing speed of the CPU and the relatively slow operating speed of I/O devices. By allowing two or more programs to be executed concurrently, one program may be performing I/O operations, while the CPU is free to be used by other programs.
- c. Times-sharing allows many people to use the system simultaneously.
- (1) Many users can be served at one time by the CPU because of the vast difference in the time used to transmit or receive data and the time taken to execute instruction.
  - (2) By using terminals, users may access the computer from remote locations. Because data travel from the terminal to the computer at the speed of light, a user in New York can get an answer from a computer in Los Angeles within two seconds.
  - (3) User's response time must be limited to two or three seconds.
  - (4) Time-sharing operating systems are more complex than batch systems, because they require far more I/O activity.
- d. Real-time systems have the same general characteristics as time-sharing systems. However, with the real-time system the data is entered, processed, and output in time to affect the transaction or business at hand.
- (1) Generally one program serves all the users; whereas, in time-sharing, users may call in their programs from either a private or common library.
  - (2) Most real-time systems can run batch jobs at the same time as real-time applications are run.
  - (3) Normally, the processing time of a program is much shorter than the time that is actually required by that program to run. The processing speed of the CPU is much greater than the speed at which the I/O devices can operate; therefore, it takes a great amount of time for the computer to get its input and to produce its output.

- e. In a sample, a real-time system assumes that two problem programs are in storage for concurrent execution. The supervisor would then ensure efficient processing as follows:
  - (1) One program receives control of the CPU and can remain active until it has to issue an I/O request.
  - (2) Supervisor takes control, starts the necessary I/O operation, and gives control to the second program, which can now use the CPU while the first program waits to complete the I/O process.
  - (3) When the I/O operations are complete, the first program is given control of the CPU again.
  - (4) This method ensures that the CPU is idle only if both programs have to wait for I/O, that is, if the second program issues an I/O request before the I/O operation for the first program is complete.
  - (5) This idle time, however, is much shorter than it would be if the two programs were executed one after the other.
- f. A prerequisite of multiprogramming is that storage is divided into sections.
  - (1) In DOS/VS these are referred to as partitions.
    - (a) Each partition can contain only one program at a time.
    - (b) The number of partitions in a system, as well as their sizes, is defined at the time when the system is generated.
    - (c) You cannot change the number of partitions during system operation, but you can modify their size by means of an operator command.
    - (d) You have one background partition (BG) and up to four foreground partitions (F1, F2, F3, F4). The background partition is always present.
  - (2) Priority rules the processing order, because the programs in a multiprogramming environment take turns with each other in using the CPU.
    - (a) Processing priority of a program depends on the priority of the partition in which it runs.

- (b) Normally, F1 has the highest priority, followed by F2, F3, and F4; the background has the lowest priority.
    - (c) These priorities are default values, they are valid when nothing else has been specified.
  - (3) In single-partition systems (BG only), only one program can be in storage at a time. If it needs input or output, it issues an I/O request to the supervisor, which initiates that request and after completion, gives control back to the problem program.
- B. **Language translators** are programs that translate the English-like program instructions of a high-level language such as COBOL and BASIC into machine-readable language.
  - 1. **Compilers** are translators that translate the entire source code (the program) into object code (machine language).
    - a. The program cannot be executed until the translation is error-free, called a clean compile--this means there are no diagnostic messages. At this time, the program is error-free; that is, it does not contain any syntax errors.
    - b. Logic errors are discovered through debugging, or running the program against test data.
    - c. COBOL is an example of a compiled language.
  - 2. **Interpreters** translate the source code one line at a time as it is being executed.
    - a. One line is translated and executed, then the next line is translated and executed.
    - b. Syntax errors are found as the program is run. This method is distinctly slower, as the same code in loops will be continuously translated and executed.
    - c. BASIC is an example of an interpreted language.
  - 3. **Assemblers** translate assembler language into machine code. It translates the entire code at one time.
- C. **Utilities** are programs that have been written to accomplish common tasks such as sorting records, copying files, or formatting disks. This enables programmers to increase their productivity by using utilities directly, or by incorporating them into their applications, rather than writing their own code for these tasks.



- D. The user interface shell is generally like a menu. It offers an interface between the user and the operating system. Options may be selected from a menu, thus relieving the user from the task of learning and remembering complex operating system commands. It also protects the operating system from the user.
- E. In virtual storage, programmers do not have to write their programs in such a way that they fit entirely into the computer's storage. They are written for a much larger storage, which does not really exist. Therefore, the storage is called virtual storage (as opposed to the computer's real storage).
  - 1. Virtual storage consists of two separate areas:
    - a. The real address area covers all the addresses that exist in the real storage of the computer.
    - b. The virtual address area starts just beyond the highest address of the real address area and ranges up to a value that was specified when the DOS/VS system was generated. Virtual storage is normally considerably larger than real storage.
  - 2. Virtual storage is divided into a number of partitions and introduces the concept of "paired" partitions.
    - a. Each pair has one part in the virtual address area; this is called the "virtual partition."
    - b. The other part is in the real address area and is called the "real partition." The virtual and real partitions to which no storage has been allocated are set to 0K.
  - 3. A 3-partition DOS/VS system in which storage has been allocated to the virtual partitions BGV and F1V and to the real partition BGR appears as follows:
    - a. The supervisor always resides in the lower part of the real address area.
    - b. No storage has been allocated to F2R and F1R; both are set to 0K.
    - c. No storage has been allocated to F2R; it is set to 0K.
  - 4. A program to be executed under DOS/VS must run either in a virtual partition (virtual-mode execution) or in a real partition (real-mode execution); it cannot use a combination of both. If, in the above example, a program runs in BGV; no program can run in BGR and vice versa.

5. The execution of a program may appear to take place in virtual storage. However, programs actually require real storage. This is because the virtual address area does not have physical space in real storage that is directly related to it.
6. Page data set is when the system places all of the virtual programs on an intermediate storage device, as an area on a disk pack as an intermediate step.
7. The disk pack resides on the disk drive assigned to SYSRES.
  - a. The page data set is associated with the virtual address area in the same way as real storage is associated with the real address area.
  - b. The programs on the page data set also require real storage.
8. DOS/VS solves this problem by making all the real storage that is not being used by the supervisor and real-mode programs available to the programs that reside on the page data set. The collection of all unused sections of real storage may be thought of as a pool on which all programs on the page data can draw.
9. Virtual storage is normally much larger than real storage. Consequently, not all the virtual programs can be accommodated by real storage at the same time.
10. Real storage is divided in 2K blocks called page frames.

## II. Types of operating systems

### A. Microcomputer

1. MS-DOS was developed by Bill Gates at Microsoft Corporation and is the primary operating system for IBM microcomputers.
  - a. Versions of MS-DOS
    - (1) MS-DOS 1.0 was developed in August 1981, for the IBM PC. It has features which include: using single-sided disks with eight sectors per track. While they could access up to 64K at that time, 64K was considered a huge amount of storage space.
    - (2) New features of MS-DOS Version 1.1 included: uses double-sided disk with 40 tracks per side, or a total of 80 tracks; 320K of storage; and an access to 640K of RAM.



- (3) MS-DOS Version 2.0 was developed in 1982. With the introduction of the hard disk, the file organization was restructured using a system of subdirectories. The primary feature was that it provided support for the hard disk for the IBM XT.
- (4) MS-DOS Version 2.1 was released in 1983 with the introduction of the IBM PC jr, and it provided an additional feature of 1/2" height disk drive support.
- (5) MS-DOS 3.0 was released in 1984 with the introduction of the IBM PC AT. It used high-quad-density diskettes with 15 sectors per track, and 80 tracks per side. In addition, it has 1.2MB of storage, access to 640K of RAM, and networking capabilities.
- (6) MS-DOS 3.2 was released in 1985, and features enhanced capabilities, debugged.
- (7) MS-DOS 3.3 was released in 1987. It introduced the PS/2 family. It supports 3.5-inch disk and provides 720K or 1.44MB of storage.
- (8) MS-DOS 4.01--512 MEG C drive. Has a DOS shell to facilitate use of DOS commands. Allows a hard drive bigger than 30 MEG. Latest DOS at this time.

b. MS-DOS disk configuration

- (1) Sector usage
  - (a) Boot area sector 0 is also called the boot record. It consists of a short machine-language program that starts the process of loading DOS into memory.
  - (b) The FAT area (File Allocation Table) is located in sectors 1-4--keeps a record of where files are stored and how space is organized on the disk.
  - (c) The root directory area is sectors 5-11. This area contains information on how files are organized in the root directory. It is a fixed size of 112 entries. When hard disks were developed, subdirectories were developed to overcome the limitation of root directories.
  - (d) The data area is sectors 12-719. This area is the place where data in files are stored.

- (2) The DOS 3.2 disk specifications
  - (a) Double-sided, double-density, soft-sectored.
  - (b) Has 9 disk sectors per track.
  - (c) Has 40 tracks per side; total of 80 tracks.
  - (d) Has 719 sectors.
  - (e) Has 354 clusters for data files.
  - (f) One cluster = 1024 bytes = 2 sectors. The space for files is allocated in clusters. The smallest amount of space the computer writes at one time is one cluster. 1 sector = 512 bytes, and 1 cylinder = the same number track on both sides of disk.

c. MS-DOS system files

- (1) IBMBIO.COM is a read-only, hidden-system archive file with 9,564 bytes, 10 clusters, and starts at sector 12, cluster 2.
- (2) IBMDOS.COM is a read-only, hidden-system archive file with 27,760 bytes, 28 clusters, and starts at sector 32, cluster 12.
- (3) COMMAND.COM is an archive file, the only system file that is displayed when you list the files on the disk. It has 23,210 bytes, 23 clusters, starts at sector 88, cluster 40.

\* **MS-DOS SYSTEM FILE NOTES:** When you attempt to put the operating system on a disk using the SYS command while there is information already there, you will often get the message that there is not enough room on the disk to transfer the system. That is because MS-DOS tries to put the system in one place on the disk, and quite often that area is already used. There are some utility programs that will move the data files to another place on the disk, thus allowing the storage of the operating system files.

- 2. The OS/2 operating system was designed by Microsoft in cooperation with IBM for its new family of PS/2 computers. It is considered the state-of-the-art operating system for managing information on the personal computer.

- a. It allows computers to run several programs at the same time.
  - b. It allows computers to handle bigger programs and access more stored memory than can be handled by its predecessor, MS-DOS.
  - c. The OS/2 operating system will supersede MS-DOS as the operating system of choice by the end of 1990, with an estimated 3.7 million copies shipped that year vs. 1.5 million copies of MS-DOS.
  - d. The OS/2 standard version will be compatible with 80286- and 80386-based computers. By utilizing the advanced capabilities of the 80286 and 80386 microprocessors, OS/2 provides the platform for advanced software applications offering a whole new class of productivity to users through its multitasking and larger memory addressing capability.
  - e. Versions of OS/2
    - (1) The OS/2 Standard Edition 1.0 was released in December 1987. It supports enhanced memory space, multitasking and the existing MS-DOS environment, can access up to 16MB of RAM directly, and can run many MS-DOS programs in the 8088 mode.
    - (2) The OS/2 Standard Edition 1.1 adds IBM's Presentation Manager, a graphics-based program interfaced with windowing capabilities.
    - (3) The OS/2 Extended Edition offers a relational database system, terminal emulation, and connectivity. It consists of application development facilities, remote/local-area network data services, Systems Network Architecture (SNA/LAN gateway support), and IBM's Enhanced Connectivity facility.
3. APPLE IIe DOS 3.3
- a. When you format an Apple DOS 3.3 disk, you generally create a bootable disk. That means that the Apple DOS is installed on the disk. To format an Apple disk, you key in the command: INIT followed by a name, usually HELLO. This is the name of the program that will be run when the computer is booted. When you format an Apple DOS 3.3 disk, the following things happen:
    - (1) All previous information is erased from the disk.

- (2) The 35 tracks are set up on one side of the disk. The Apple uses the backside of the disk. The other side of the disk may be used, but a write-enable notch must be created on the other side of the disk.
  - (3) Each track is divided into 16 sectors, and each sector contains 256 bytes for a total capacity of 163,840 bytes.
  - (4) The Apple Disk Operating System 3.3 is stored on tracks 0, 1, and 2.
  - (5) The index is created on track 17. The index track keeps track of the file type, file name, and the beginning location of the file.
- b. The Apple DOS 3.3 writes one sector (256 bytes) at a time to the disk. As DOS writes files to disk, one sector at a time, the sectors may not be continuous. At the end of each sector, there is a pointer which points to the next sector for that file.
4. APPLE DOS 1.1 professional disk: ProDOS is faster, easier to use, and more powerful than the DOS 3.3.
- a. With the introduction of hard disks with a limitation of 105 files on the index track, there was a need for a new file organization to use the full capacities of the hard disk.
  - b. The new file organization created a tree structure with one root directory and many subdirectories. Each subdirectory may also have up to 64 levels of subdirectories.
  - c. To access these subdirectories, you create path names. This is also the reason MS-DOS upgraded to version 2.0. Previous versions prior to the introduction of the hard disk limited files to 116, the number of available entries in the directory.
  - d. The ProDOS operating system is generally stored on a startup disk. The computer is booted with this startup disk, then the disk is removed. (This is similar to MS-DOS system disks.) The system disk contains:
    - (1) Several utilities.
    - (2) A language processor.
    - (3) A user interface shell to guide users through the necessary file and disk handling utilities.

- e. Unlike DOS 3.3, most disks for use under ProDOS do not have the operating system on them. The operating system requires too much disk space, as does the MS-DOS system disk. Therefore, a startup disk is used to load the operating system, then removed for normal use.
  - f. The general organization of the ProDOS disk contains 35 tracks, 8 blocks per track, 2 sectors per block, and 512 bytes. ProDOS writes one block at a time to disk and uses a single-sided disk.
5. **MAC--**The operating system is written exclusively by Macintosh. This operating system is meant to be completely invisible to the user. The system is user-friendly; just point and click.
- a. Finder file version 4.1 will run on the Macintosh 512K and the Macintosh 512Ke.
  - b. System file version 3.0 and version 3.2 will run on the Macintosh 512K, the 512Ke, and the Plus.
  - c. System file version 4.0 and version 4.01 will run on the Macintosh 512K, the 512Ke, the Plus, the SE, and the II.
  - d. System file version 4.2 and 6.0 will run on the Macintosh Plus, the SE and the II.
  - e. System 7 has recently been announced and will soon be available.
- B. **Mainframe operating systems:** In the early days, management of the installation's resources (such as CPU, input/output devices, programs,) was fairly easy. However, as installations grew, a way of simplifying resource management became necessary. Special programs were written to perform operations that occur frequently, and these programs were combined to form an operating system. Because the operating system resides on disk and uses the virtual storage concept, it is called the Disk Operating System/Virtual Storage (DOS/VS). DOS/VS can be divided into two main programs:
- 1. Control programs ensure proper and efficient operation of the entire system. The three control programs are:
    - a. The Initial Program Loader (IPL), the DOS/VS IPL program which starts the system. This means that this program must be executed each time the system is started. Executing the IPL program is the responsibility of the operator.

- b. The supervisor supervises execution of all the other programs. It coordinates the use of the system resources and maintains the flow of operations, thus taking over a considerable part of the burden which otherwise would have to be handled by the individual programs.
  - (1) Some of its functions are to load programs; manage and control storage and CPU time; and to handle much of the communication between the system and operator.
  - (2) When the IPL procedure has been performed, part of the supervisor will always be in storage. The remainder of the supervisor comprises parts that are less frequently used; they are called "transient routines."
  - (3) Transient routines are kept on the disk pack on which the system resides. Then they are automatically brought into the so-called transient area in storage when they are required.
- c. Job control
  - (1) The job control program is loaded into storage each time a new program must be prepared for execution.
    - (a) This means that it is loaded for the first time after IPL to prepare execution of the first program.
    - (b) Subsequently it is loaded each time a program has completed execution or is terminated. This makes transition from executing one program to executing the next one automatic.
  - (2) The job control program has many different functions. Not all of them are needed for every program execution. The programmer or sometimes the operator must therefore inform the job control program of the specific requirements for each program. Such instructions to the job control program, which have a prescribed form, are called "job control statements" and "job control commands." The difference between these two is called the "job control language."
- 2. Operator to system communication is done by means of either job control statements or operator commands. Job control statements have two slashes (//) in positions 1 and 2; whereas, operator commands do not have slashes (their operation code may, but need not, start in position one).



- a. Job control statements may either be punched in cards and entered by a card reader (SYSRDR) or be in card-image format on tape, disk, or diskette. They are accepted by the system only between jobs or job steps.
  - b. There are five types of operator commands:
    - (1) IPL commands are accepted only at IPL time (when you perform the IPL procedure). These commands prepare the DOS/VS system for operation.
    - (2) Attention commands are entered and accepted after you have pressed request on the console keyboard.
    - (3) Job control commands are accepted only between jobs or job steps.
    - (4) POWER/VS commands are treated like attention commands.
    - (5) VTAM network operator commands are needed to monitor and control a VTAM telecommunication network.
3. Processing programs are made up of language translators, service and utility programs, and problem programs, also called "application programs." The first two types of processing programs are supplied by IBM. They serve to make the application programs run under DOS/VS and provide service functions to make the best use of the system.
- a. Language translators translate source programs. DOS/VS permits the programmer to write his programs in several programming languages: COBOL, PL/1, FORTRAN, Assembler, and RPG.
    - (1) A program written in one of these languages is called a source program.
    - (2) Source programs must be translated into a language that can be understood by the computer.
    - (3) Translation of source programs is performed by special programs called "language translators."
    - (4) Each programming language has its own language translator. In the case of the assembler language, the language translator is called the "assembler;" the translators of the other programming languages are called "compilers."

- (5) The translation of a source program results in an object program which can be punched or written out, or stored (cataloged) in the library that contains object modules.
  - (6) Object modules cannot yet be executed. They must first be processed by the linkage editor.
- b. **Service and system utility programs:** The DOS/VS service program consists of the linkage editor program and a set of programs called the "librarian." The DOS/VS system utility programs are IBM-supplied programs designed to perform routine tasks.
- (1) Linkage editor reads the output from a language translator and converts it to an executable program. This conversion is called "link-editing." The linkage editor is further used to catalog executable programs, or parts of these programs (called "phase") in the library that contains executable programs (core-image library).
  - (2) The librarian consists of a number of programs that are used to maintain the DOS/VS libraries. The librarian performs such functions as: displaying the contents of a library, adding elements (phases, modules, etc.) to a library, deleting elements from a library, condensing a library, and creating a library and utility programs.
  - (3) Utility programs are designed to perform such everyday tasks as copying data from one volume to another and preparing volumes for the storage of data.
    - (a) The Assign Alternate Track Data Cell program analyzes a data-cell track that contains an error and assigns an alternate track to contain the data from the defective track.
    - (b) The Analysis Program-1 (AP-1) enables the operator or system programmer to determine if a data error on the IBM 3344 or 3350 Direct-Access Storage Device is a data module error or a device error.
    - (c) The Assign Alternate Track Disk program analyzes a disk track that contains an error and assigns an alternate track to contain the data from the defective track.
    - (d) Backup and Restore System programs generate on tape a back-up copy, independent of the system and/or private libraries, and restores this copy to another DASD device.



- (e) Clear Disk and Data Cell programs clear and preformat one or more areas of a disk pack or a data cell.
  - (f) The Copy and Restore Disk or Data Cell program copies a file or volume from disk to card, disk, or tape, or from a data cell to tape. It also restores data to disk or data cell.
  - (g) The Copy and Restore Diskette program copies the contents of a diskette to another diskette on a different device. It also restores data from one diskette to another diskette on the same device.
  - (h) The Copy File and Maintain Object Module program copies the contents of a diskette to another diskette on a different device. It also restores data from one diskette to another diskette on the same device.
  - (i) The Copy File and Maintain Object Module is a multipurpose program which serves primarily as a file-to-file utility used to copy and list card image files. It may also be used to modify object modules, phases, and PTFs as well as to deblock PTF files.
- (4) **DOS/VS libraries** contain programs or program parts as well as control information. Their contents are available for use whenever they are required for processing. There are four types of libraries:
- (a) **Core-image** library or phases contains program phases that have been processed by the linkage editor and are therefore ready to be executed. A phase can be an entire program or part of a program. Phases are placed into the core image library by the linkage editor.
  - (b) **Relocatable** library contains object modules. The object modules can be combined with other object modules and processed by the linkage editor to form a single executable program. The relocatable library is used mainly to store standard routines that are frequently used in larger programs. This makes it unnecessary for the programmer to code the same routine more than once for several programs.

- (c) **Source** statement library contains source programs or parts of such programs (source statements or groups of source statements). The individual elements contained in a source statement library are called "books." Source statements are frequently used in larger source programs.
  - (d) **Procedure** libraries are used to store frequently used sets of job control and linkage editor control statements, as well as control statements for system utility and service programs.
  - (e) **System/Private** libraries of a library are available as system and as private libraries. The procedure library is available as a system library only. All system libraries together form the DOS/VS system. They all reside on the same disk pack, which is referred to as the "system residence pack," or SYSRES.
  - (f) Library directories precede each system library. They contain the names of all elements in the respective library and the disk addresses where they can be found.
- III. **Application software**--prepacked software written by professional programmers and sold for use of less experienced persons.
- A. Word processing programs deal with the processing of words for letters, research papers, books, etc. Some examples are MicroSoft Word, PFS Professional Write, WordPerfect, and Wordstar.
  - B. Spreadsheets are programs that perform mathematical calculations on columns and rows of numbers. They are usually used for business accounting. Of all the business-related programs available, spreadsheets probably have the widest use. These were available on micros before they became available for larger systems. Some examples of popular business spreadsheet packages include Lotus 1-2-3, Excel, Multiplan, and Quattro.
  - C. Databases are programs designed to organize and sort lists of information. The sophistication of many of the database programs now available makes them practical for both business and home use. Some examples of popular business database packages include dBASE IV, Reflex, and R:BASE.
  - D. Graphics programs such as paint, drawing, and desktop publishing programs are designed specifically for the graphic representations of data. There has been no area of development which has advanced more than graphics. Some examples include; Deluxe Pain 2000, AutoCAD, Xerox Ventura, Harvard Graphics, and PageMaker.

- E. Communications programs allow for communication between computers over phone lines.
- F. Integrated packages offer several programs in one bundle--usually a word processor, a spreadsheet, a database and graphics or communication programs. Examples include Microsoft Works, Appleworks, and Symphony.
- G. Utilities are programs designed to enhance and improve PC operations. Examples include Norton Utilities and Tornado. Many are available as public domain disks.
- H. Programming languages are available for the microcomputer. These include: BASIC, Pascal, Macro assembler, COBOL, and FORTRAN. These programs include compilers and are very good for education and general purpose programming.

## ACTIVITY 1: CHARACTERISTICS AND FUNCTIONS OF AN OPERATING SYSTEM

1. Describe the Input/Output Control System.
2. Describe the three basic functions of an operating system.
  - 1.
  - 2.
  - 3.
3. Explain the overall functions of the operating system.

## ACTIVITY 2: BATCH AND REAL-TIME PROCESSING

1. Discuss the various applications using batch processing.
2. On a separate sheet of paper, diagram the typical configuration of batch processing.
3. On a separate sheet of paper, diagram the typical multiprogramming environment.
4. Evaluate and list the typical real-time systems.
5. On a separate sheet of paper, diagram the typical mainframe operating system components.

### ACTIVITY 3: COMPILERS, INTERPRETERS AND VIRTUAL STORAGE

1. Write a paragraph comparing compilers and interpreters.
2. Read current literature, and write a brief summary on a compiler or an interpreter. Check with your teacher to make sure that you are not writing on the same topic as your classmate.
3. Read current literature, and write a summary on a mainframe or a microcomputer utility.
4. Explain the two areas of virtual storage.
  - 1.
  - 2.
5. Diagram the concept of virtual storage.
6. Diagram the use of partitions and their advantages.

#### ACTIVITY 4: MICROCOMPUTER OPERATING SYSTEMS

1. Chart the development of the Microsoft Disk Operating System (MS-DOS).
2. Read articles about Bill Gates, and write a paragraph about his accomplishments.
3. Experiment with and explain what the operating system does when:
  - a. Access is attempted with an initialized disk.
  - b. Access is attempted to a different brand of computer.
4. Practice the steps of initializing disks both with and without operating system components.
5. Using a DOS reference manual or guide, outline the functions of DOS for a personal computer.

## ACTIVITY 5: COMPONENTS OF THE DOS/VS OPERATING SYSTEM

1. List and describe the three control programs.

- 1.

- 2.

- 3.

2. List the five types of operator commands.

- 1.

- 2.

- 3.

- 4.

- 5.

3. List the three types of processing programs.

- 1.

- 2.

- 3.



### ACTIVITY 5 (Continued)

4. List the functions performed by the librarian.
  - 1.
  - 2.
  - 3.
  - 4.
  - 5.
5. List the four types of DOS/VS libraries.
  - 1.
  - 2.
  - 3.
  - 4.
6. Compile a generic list of mainframe operating system components.
7. Describe DOS functions not common to mainframe operating systems.

## SECTION II: COMPUTER HARDWARE UTILIZATION

### UNIT III: COMPUTER OPERATION AND MAINTENANCE

#### UNIT

**OBJECTIVE:** The students' knowledge will be reinforced in all phases of operating computer hardware.

#### STUDENT

**COMPETENCIES:** Upon completion of this unit, the student will be able to:

1. Identify terminology related to computer operation and maintenance.
2. Operate and maintain machines.
3. Monitor and report environmental problems.
4. Inventory resources and report findings.
5. Handle media and support tools.
6. Monitor hardware performance.
7. Schedule workflow.
8. Maintain data and physical security.

**FOCUS:** Being familiar with all phases of computer hardware operation is valuable in most businesses.

## TERMINOLOGY

|                                      |                                                                                                                                                                                                                                                          |
|--------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Application program</b>           | Software used to execute the procedures within a system. Programs that run on the computer to perform a particular application.                                                                                                                          |
| <b>Arithmetic and logicity (ALU)</b> | The ALU consists of registers and logic circuitry and is responsible for performing mathematical operations.                                                                                                                                             |
| <b>Batch processing</b>              | The serial execution of programs. Each program is entered into the CPU and is completely executed before a second program is executed.                                                                                                                   |
| <b>Book</b>                          | The term denotes the source statements, or data, cataloged under a given name.                                                                                                                                                                           |
| <b>Boot area</b>                     | On the MS-DOS disk (also called "boot record"); consists primarily of a short, machine-language program that starts the process of loading DOS into memory. If it does not find the system files on the disk, it displays the message "Non System Disk." |
| <b>Buffer</b>                        | An area reserved for data either coming into or leaving main memory. The data is stored in the buffer until the information can be processed or, in the case of the output buffer, until the output device is ready for the data.                        |
| <b>Central control unit (CCU)</b>    | The CCU is responsible for fetching an instruction, breaking the instruction into parts, moving the necessary data into the ALU, and seeing that the instruction is executed.                                                                            |
| <b>Central processing unit (CPU)</b> | The mainframe of the computer contains the CPU, ALU, main memory, input control unit, and output control unit. Sometimes the term CPU is used to denote the computational ability of the computer rather than the five components of the mainframe.      |
| <b>Channel</b>                       | A physical device that transmits data from an I/O device to the CPU or from the CPU to an I/O device.                                                                                                                                                    |
| <b>Compiler</b>                      | A program used to translate high-level language statements written in a programming language such as COBOL into machine-language instructions.                                                                                                           |
| <b>Computer operator</b>             | The individual responsible for setting up the computer system to execute jobs, responding to messages generated, and monitoring the computer system in order to achieve maximum throughput.                                                              |

|                            |                                                                                                                                                                                                                                                                                                                                 |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Computer system</b>     | A computer system, often called a system, consists of the CPU, I/O devices that are used to input and output data, and the operating system.                                                                                                                                                                                    |
| <b>Controller</b>          | The hardware that coordinates the movement of data between the CPU and an I/O device.                                                                                                                                                                                                                                           |
| <b>Control unit</b>        | Hardware that includes controlling circuitry for the attached input or output devices (the controller).                                                                                                                                                                                                                         |
| <b>Core dump</b>           | A printout of the computer's memory contents. Core dumps are sometimes used in debugging programs.                                                                                                                                                                                                                              |
| <b>Core image library</b>  | A portion of the SYSRES pack reserved for storing ready-to-execute programs such as compilers and application software.                                                                                                                                                                                                         |
| <b>Defacto standard</b>    | The standard that vendors informally accept, generally because it has come to dominate a business market segment.                                                                                                                                                                                                               |
| <b>Dumb terminal</b>       | A terminal that cannot be programmed. It is always under the direct control of the host computer. It can do no processing on its own.                                                                                                                                                                                           |
| <b>Event-driven</b>        | A basis for multiprogramming when the CPU processes a program until it requires an I/O operation.                                                                                                                                                                                                                               |
| <b>Family of computers</b> | A group of computers that have the same general architecture. At the low end of the family, a limited amount of memory and slow I/O devices are available. At the high end of the family, the I/O devices are much faster, more memory can be obtained, and a wider range of software products is usually available.            |
| <b>FAT area</b>            | File Allocation Table keeps a record of where files are stored and how space is used on the MS-DOS disk.                                                                                                                                                                                                                        |
| <b>Foreground</b>          | An area within the CPU's memory. Some systems' real memory is divided into areas referenced as F1, F2, F3, F4, and BG (background). When a message is displayed pertaining to the execution of a job, the system usually indicates the area (or partition) in which the job is running.                                         |
| <b>Front-end processor</b> | An interface between the terminals and the computers. The front-end processor, or controller, may only assemble the bits transmitted into characters and provide a parity check. Other front-end processors are minicomputers or microcomputers and assist the host computer with routine functions required in teleprocessing. |

|                                           |                                                                                                                                                                                                                                                  |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>High-level language</b>                | A language, such as COBOL, which permits the programmer to write source code in English-like statements. The compiler translates each source code instruction to one or more machine-language instructions.                                      |
| <b>Input/output control system (IOCS)</b> | A series of commands for initiating, executing, monitoring, and controlling data transfer into and out of the computer.                                                                                                                          |
| <b>Input/output (I/O)</b>                 | Computer hardware designed to enter data devices into a computer system and to receive the results of the processing of data.                                                                                                                    |
| <b>Interactive computing</b>              | Direct communication between the terminal user and the computer.                                                                                                                                                                                 |
| <b>Interpreter</b>                        | A program which checks each source code statement for errors and immediately translates the statement to object code and executes it.                                                                                                            |
| <b>Interrupt</b>                          | Due to a normal or an abnormal occurrence, control of the system's resources returns to the supervisor.                                                                                                                                          |
| <b>Interrupt</b>                          | A basis for multiprogramming when the CPU processes a program until it requires an I/O operations.                                                                                                                                               |
| <b>IPL</b>                                | Initial Program Load. Indicators and internal switches are reset, and the supervisor is loaded into the CPU. When a system is IPLed, the operator must depress certain keys on the console and then enter the current date and time.             |
| <b>IPO switch</b>                         | Immediate Power Off switch. A switch used in the case of an emergency to cut off the computer's power.                                                                                                                                           |
| <b>Job</b>                                | A procedure initiated by a control card or a command entered through a terminal or console typewriter. A job can consist of one or several tasks. The end of a job is usually indicated by the use of a JCL card such as the /& card.            |
| <b>Job control language (JCL)</b>         | A specialized language for setting up jobs and communicating basic instructions to the computer.                                                                                                                                                 |
| <b>Job entry control language</b>         | Control statements used under a software feature such as POWER to provide additional options for the queues.                                                                                                                                     |
| <b>Library</b>                            | A collection of programs, routines, and, in some cases, source code stored on magnetic disk.                                                                                                                                                     |
| <b>Linkage editor</b>                     | A program supplied by the vendor as part of the operating system that combines the user's object modules with modules from the core-image library. The linking of the user's modules with those from the library produces an executable program. |

|                            |                                                                                                                                                                                                                                                                                            |
|----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Machine language</b>    | The absolute, or actual, instructions executed by the computer in reading, processing, and writing data.                                                                                                                                                                                   |
| <b>Memory</b>              | A type of storage used to store programs or data that exist on an auxiliary device.                                                                                                                                                                                                        |
| <b>Memory management</b>   | The allocation of memory.                                                                                                                                                                                                                                                                  |
| <b>Microcomputer</b>       | A computer containing a CPU on a single chip. It is a set of Large-Scale Integration (LSI) circuits capable of performing the essential function of a computer. Within a microcomputer, other chips and circuit boards are used to control I/O activities and to add memory to the system. |
| <b>Multiplexer channel</b> | A connection between the control unit and the computer. Data from two or more I/O devices are interleaved a byte, or a block, at a time while being transferred between the control unit and the computer.                                                                                 |
| <b>Multiprocessing</b>     | Two or more CPUs are combined into a single system so that more than one instruction can be executed at one time.                                                                                                                                                                          |
| <b>Multiprogramming</b>    | Two or more computer programs are executed concurrently by a single computer.                                                                                                                                                                                                              |
| <b>Multitasking</b>        | Operating systems that allow concurrent tasks.                                                                                                                                                                                                                                             |
| <b>Node</b>                | A computer that provides the interface between terminals and the software supporting communication devices and terminals.                                                                                                                                                                  |
| <b>Object code</b>         | Machine-language instructions. After a program has been written, compiled, and debugged, the object code version of the program is often stored in an on-line library.                                                                                                                     |
| <b>Off-line</b>            | A data processing operation or device that is not under the direct control of the computer.                                                                                                                                                                                                |
| <b>On-line</b>             | An operation or device under the direct control of the computer.                                                                                                                                                                                                                           |
| <b>Operating system</b>    | Software that controls the execution of computer programs. It may provide for scheduling, debugging, I/O control, library management, error routines, and many other related services. It runs the computer as compared to application programs that run on the computer.                  |
| <b>Operations</b>          | The area within the data processing department that involves data entry, computer and peripheral operations, job scheduling, library maintenance and management, and many other related services.                                                                                          |



|                             |                                                                                                                                                                                                                                                                                                                                 |
|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Overlays</b>             | During execution of a program, one section of real memory is reserved for storing program segments. When a section of instructions is executed, a new section is retrieved from the library and overlaid on the section just executed. Overlaying permits a program that requires more memory than is available to be executed. |
| <b>Page</b>                 | An addressable portion of a program or data.                                                                                                                                                                                                                                                                                    |
| <b>Page frame</b>           | An area in real memory used to store one page of instruction or data. Usually 2K or 4K.                                                                                                                                                                                                                                         |
| <b>Page-in</b>              | A page of instructions or data is brought into real memory from its location in virtual memory.                                                                                                                                                                                                                                 |
| <b>Page-out</b>             | Restoring a page of data or instructions in real memory. The page frame of real memory is then released and made available for paging-in a new page.                                                                                                                                                                            |
| <b>Peripheral equipment</b> | Input or output devices used in electronic data processing.                                                                                                                                                                                                                                                                     |
| <b>Polling</b>              | A method for inquiring whether a terminal has data to be transmitted.                                                                                                                                                                                                                                                           |
| <b>Port</b>                 | An entry channel to which a dataset is attached. One port is assigned to each user of the system.                                                                                                                                                                                                                               |
| <b>POWER</b>                | Priority Output Writer Execution Reader. A software package added to some operating systems that provides for spooling and multiprogramming.                                                                                                                                                                                    |
| <b>Priority system</b>      | The method used to determine which program waiting for execution should be run.                                                                                                                                                                                                                                                 |
| <b>Problem state</b>        | The CPU is executing instructions which are part of an application program.                                                                                                                                                                                                                                                     |
| <b>PSW</b>                  | Program Status Word. The PSW contains, along with other information, the operational code of the instruction being executed and the address of the next instruction.                                                                                                                                                            |
| <b>Query language</b>       | A programming language that permits the user to communicate with the computer in English.                                                                                                                                                                                                                                       |
| <b>Queue</b>                | A waiting line file used for storing either data or programs.                                                                                                                                                                                                                                                                   |
| <b>Real memory</b>          | Storage for data and programs that exists within the central processor's main memory.                                                                                                                                                                                                                                           |

|                            |                                                                                                                                                                                                                                                                                                                                          |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Real-time</b>           | Data entered into the system is processed in time to affect the activity at hand. Multiple users can simultaneously access the system by means of terminals.                                                                                                                                                                             |
| <b>Register</b>            | A temporary storage device used for special functions such as storing the instructions being executed and the address of the next instruction. Registers are also used for the arithmetic and logic operations.                                                                                                                          |
| <b>Reliability</b>         | The percentage of time that the computer is up relative to the time it is scheduled to be up and running.                                                                                                                                                                                                                                |
| <b>Relocatable library</b> | A library stored on SYSRES which contains object-code subroutines that can be incorporated into other programs.                                                                                                                                                                                                                          |
| <b>Relocatable phase</b>   | At the time of execution, the program module (or phase) can be loaded into any memory partition.                                                                                                                                                                                                                                         |
| <b>Response time</b>       | The time from when the user issues an inquiry until the computer responds with the answer.                                                                                                                                                                                                                                               |
| <b>Root directory area</b> | On the MS-DOS disk, the root directory contains basic information about the files stored in the root directory.                                                                                                                                                                                                                          |
| <b>SCP</b>                 | Systems Control Program. Another term used to denote operating system software.                                                                                                                                                                                                                                                          |
| <b>Selector channel</b>    | A connection between the control unit and one type of high-speed I/O device.                                                                                                                                                                                                                                                             |
| <b>Setup deck</b>          | Job control language cards used to initiate a job and to provide required data to execute a job.                                                                                                                                                                                                                                         |
| <b>Smart terminal</b>      | A terminal that has its own microcomputer which can be programmed. The terminal can function either on- or off-line.                                                                                                                                                                                                                     |
| <b>Source code</b>         | Computer instructions written in a pseudolanguage such as COBOL, FORTRAN, or BASIC.                                                                                                                                                                                                                                                      |
| <b>Source library</b>      | A library containing statements written in a programming language such as COBOL.                                                                                                                                                                                                                                                         |
| <b>SPOOL</b>               | Simultaneous Peripheral Operations On-Line. A technique by which slow I/O operations are accomplished faster and more efficiently by first storing the data on a storage device, usually disk, and then processing the data directly from the intermediate storage area. I/O operations may occur at the same time as program execution. |



|                         |                                                                                                                                                                                                                             |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Supervisor</b>       | Software that is sometimes called the monitor or executive program. The supervisor is stored within the CPU and coordinates all other software that make up the operating system.                                           |
| <b>SYSGEN</b>           | Generating an operating system which meets the specific needs of the user. From the options available, the user determines which ones are to be included as part of the system's software.                                  |
| <b>SYSRES</b>           | The disk pack that must be on-line for the computer system to be operational. Stored in libraries and include the user's programs, compilers, error routines, IOCS routines, linkage editor, librarian, and other software. |
| <b>System</b>           | An organized method for achieving a given result.                                                                                                                                                                           |
| <b>System commands</b>  | Instructions used by operators to communicate with a time-sharing system. System commands perform the same functions as job control language statements.                                                                    |
| <b>Task</b>             | An individual program, or routine, incorporated into a job.                                                                                                                                                                 |
| <b>Terminal</b>         | A device for communicating with the computer.                                                                                                                                                                               |
| <b>Throughput</b>       | The amount of data that, within a given period, can be entered into the system, processed, and outputted in final form.                                                                                                     |
| <b>Thrashing</b>        | An excessive amount of paging occurs during the execution of a program.                                                                                                                                                     |
| <b>Time-sharing</b>     | Multiple users, through terminals, use a single computer at what appears to be the same time. The response time should be such that each user feels that he or she has exclusive use of the system.                         |
| <b>Time-driven</b>      | The basis for multiprogramming where the CPU processes programs on a time-sharing basis, giving each program a slice of time.                                                                                               |
| <b>Transient area</b>   | An area set aside in the memory of the computer for error routines, messages to the operator, and other types of control software.                                                                                          |
| <b>Unbundled</b>        | A term denoting individual pricing of hardware, software, and support services.                                                                                                                                             |
| <b>Utility programs</b> | Frequently used programs that perform tasks such as copying files, softing files, providing backups for tapes and disks and providing tape-to-disk conversions.                                                             |

|                        |                                                                                                                                                                                                                                                                                 |
|------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Virtual memory</b>  | Storage for programs or data that exist on an auxiliary device such as a magnetic disk. Virtual memory is considered an extension of real or main memory. During execution of a program, data or parts of a program are swapped back and forth between real and virtual memory. |
| <b>Virtual storage</b> | Real memory is expanded to include system virtual memory. In a VS system, only the portion of the program executed needs to reside in real memory.                                                                                                                              |
| <b>VS</b>              | Addressable spaces on a secondary storage device (usually some type of fast-swapping memory) that appears to the programmer as real memory. From VS memory, instructions and data are swapped into real memory for execution.                                                   |
| <b>VSE</b>             | Virtual Storage Extended. Operating system software with features normally not included in a typical VS system.                                                                                                                                                                 |

## UNIT OVERVIEW

### I. Operate and maintain machines

#### A. Operating a mainframe system

1. The mainframe system has an extensive array of peripheral equipment.
  - a. Direct-access storage devices
  - b. Sequential tape drives
  - c. Printers
  - d. Mass storage devices
  - e. Local and remote terminals
2. Mainframe operation rooms require a raised floor.
  - a. Large, square panels may be lifted up.
  - b. All necessary wiring connecting the peripheral equipment to the processor is underneath the floor.
3. The operator's console allows equipment to be controlled at one location.
4. Power-up: System needs initial instructions to load the operating system into its memory.
  - a. This is called Initial Program Load, or IPL.
  - b. The initial program loading may take up to 15 minutes.
5. Power-up procedure
  - a. Depress ON button on system console keyboard. This ON button controls all peripheral devices.
  - b. When the disks reach operating speed, the ready light appears.
  - c. Tells the system where to find the SYRES disk pack, disk containing all the operating system files. The supervisor that will be used must also be identified.
  - d. Set the system time and date.

- e. Create the page data set, which is the area to be used for virtual storage.
    - f. Instruct operating system to partition memory, allocating blocks of memory to each partition.
  - 6. Power-down
    - a. Make sure all time-sharing users are logged off.
    - b. Terminate the time-sharing program.
    - c. Terminate POWER (the PEND command can be issued, which will shut down POWER).
    - d. Stop each partition.
    - e. Disengage the disk drives (do not shut off the power with the disk drives still in the ready state).
    - f. When the ready lights go off, it is now safe to shut off the power supply by pressing the off button on the keyboard console.
- B: Operator-to-system communication
  - 1. Communication with the computer system may be accomplished in two ways:
    - a. Operator commands do not have slashes (their operation code may, but need not, start in position one).
    - b. Job control statements
  - 2. Jobs and job steps
  - 3. The job control program
  - 4. Interpreting the job stream
- C. Tape drive
  - 1. Typical tape applications
    - a. Back up master files and databases.
    - b. Provide historical reference.
    - c. Back up transaction files.
    - d. Provide communication between the two systems.

- e. Provide input for an off-line printer COM subsystem.
- f. Off-load data stored in on-line disk files.
- g. Submit financial reports.
- h. Transmit software.
- 2. Tape format
  - a. Leader
  - b. Tape mark
  - c. Create tape labels
  - d. Checking tape labels
- 3. Physical and logical records
  - a. Blocking factors
  - b. IBGs
  - c. Tape density
- 4. Data representation
  - a. EBCDIC
  - b. ASCII
  - c. BCD
- 5. Operator responsibilities
  - a. Understand reasons for indicator/action pushbuttons
  - b. Observe tape handling rules.
  - c. Respond to console messages.
  - d. Clean tape unit/tapes.
  - e. Maintain tape logs and external tape labels.
  - f. Maintain tape security and prevent accidental overwrite.
  - g. Retain and store files.
- D. Disk drives

1. Types of disk subsystems
  - a. Removable
  - b. Nonremovable
  - c. Fixed-head drives
  - d. Data modules
2. File organization methods
  - a. Sequential
  - b. Random or direct
  - c. Indexed
3. Reserved areas
  - a. VTOC
  - b. Overflow for ISAM
  - c. Cylinders reserved for defective tracks
4. File security and disaster plan
5. Console messages
  - a. No label space in VTOC
  - b. No format label
  - c. No record found
  - d. Equal file in VTOC
  - e. Extent overlap on another file
  - f. Wrong pack mount, pack number
  - g. Invalid DLPL function
  - h. Invalid logical unit

E. Printers

1. Types of printers
  - a. Impact

- b. Nonimpact
- 2. Operator responsibilities
  - a. Understand reasons for indicators and action pushbuttons
  - b. Load forms.
  - c. Align forms.
  - d. Set print controls.
  - e. Respond to console messages with appropriate commands for the printer.
  - f. Remove forms.
  - g. Perform routine maintenance functions.
  - h. Clean printer.
  - i. Change ribbon and toner or ink.

F. CRT/Terminals

- 1. Kinds of terminals
  - a. Dumb
  - b. Smart
  - c. Intelligent
- 2. Data validation
- 3. Formatted screens
- 4. Conventions used with formatted screens

G. Optical scanning devices

- 1. Optical marks
- 2. Optical characters
- 3. Bar codes
- 4. Handwritten characters

H. Modems

**BCPI/Computer Hardware Utilization**  
**Unit 3: Computer Operation and Maintenance**  
**Student's Guide/UNIT OVERVIEW**

1. Types of modems
    - a. Direct connect
    - b. Acoustic coupler
  2. Modem features
    - a. Auto-answer
    - b. Auto-disconnect
    - c. Auto-dial
    - d. Auto-redial
    - e. Time-delay
- I. Microfiche
1. COM (Computer Output Microfilm)
  2. Advantages
  3. Disadvantages
- II. Monitor and report environmental problems
- A. Environment variables include: humidity, temperature, dust and dirt
- B. Hazards
1. Fire prevention and control, using sprinklers, sand, and extinguishers
  2. Priorities involving people, property, data, equipment
  3. Fire prevention relating to smoking, storage of combustibles, electrical circuits
- III. Inventory resources and report the findings
- A. Prepare forms
1. Run sheets.
  2. Maintain form supply.
    - a. Stock forms
    - b. Special forms



BCPI/Computer Hardware Utilization  
Unit 3: Computer Operation and Maintenance  
Student's Guide/UNIT OVERVIEW

- c. Quantity needed
  - d. Location
  - e. Disposition
  - f. Identification
  - g. Open boxes
- B. Inventory resources
  - 1. Terms
    - a. Quantity-on-hand
    - b. Quantity-on-order
    - c. Reorder point
    - d. Reorder quantity
    - e. Physical inventory
  - 2. Dated materials
    - a. Printer ribbons
    - b. Felt markers
    - c. Dated forms
    - d. Cleaning chemicals
    - e. Adhesive labels
- C. Organize storage according to weight, bulk, frequency of access, stability, shelf life, companion items, value
- D. Purchasing
  - 1. Selection
    - a. Catalogs
    - b. Promotional material
    - c. Prior purchases
  - 2. Processing

**BCPI/Computer Hardware Utilization**  
**Unit 3: Computer Operation and Maintenance**  
**Student's Guide/UNIT OVERVIEW**

- a. Order forms
- b. Invoices
- c. Recordkeeping
- 3. Items to purchase
  - a. Forms
  - b. Ribbons
  - c. General office supplies
  - d. Magnetic media
  - e. Containers
  - f. Maintenance supplies

**IV. Handle media support tools**

- A. Storage methods
  - 1. Media storage
  - 2. Support tools
    - a. Physical requirements
    - b. Special care needed
      - (1) Cleaning
      - (2) Avoiding magnets

- B. Diskettes
  - 1. Categories of media
    - a. Flexible diskettes
    - b. Hard disks
  - 2. What initializing does:
    - a. Formatting
    - b. Copy parts of system

- C. Libraries

1. Disk libraries include classifications, labels, recording the time and date.
  2. Maintenance libraries include scheduling, indexing and logging.
- D. Backups
1. **Terminology**--scratch and backup
  2. **Backup considerations**--the age of the media, number of copies, importance of date, and the frequency of update
  3. How to check disks
  4. Backup methods
- V. Monitor hardware performance through the console operator
- A. Contents of an error message manual for the system in use regarding:
1. Partition where message was generated.
  2. Replay priority indicator.
  3. Message identification number.
  4. Action indicator.
- B. Classification of wait states
1. Hard wait
  2. Soft wait
- C. Operator commands
1. Attention routines
  2. Screen control
- D. Interrupt processing
1. Initiate and control multiprogramming.
  2. Invoke inquiries to obtain information about status of system.
  3. Cancel the execution of a job using too many system resources or one in a loop.
  4. Use available software to determine the source of a problem.

VI. Schedule workflow

A. **Operator-run manual--documentation**

1. Narrative
2. Program error conditions
3. Run instructions

B. Types of error messages

1. No data record
2. Duplicate record
3. Invalid record code
4. Non-numeric fields of data
5. Invalid field

VII. Maintain data and physical security

A. Access to data

1. **Physical access to data--on-site and remote**
2. **Access procedures--passwords, user levels, administrator**
3. Consequences of lost data--time, money, credibility, nonreplacement

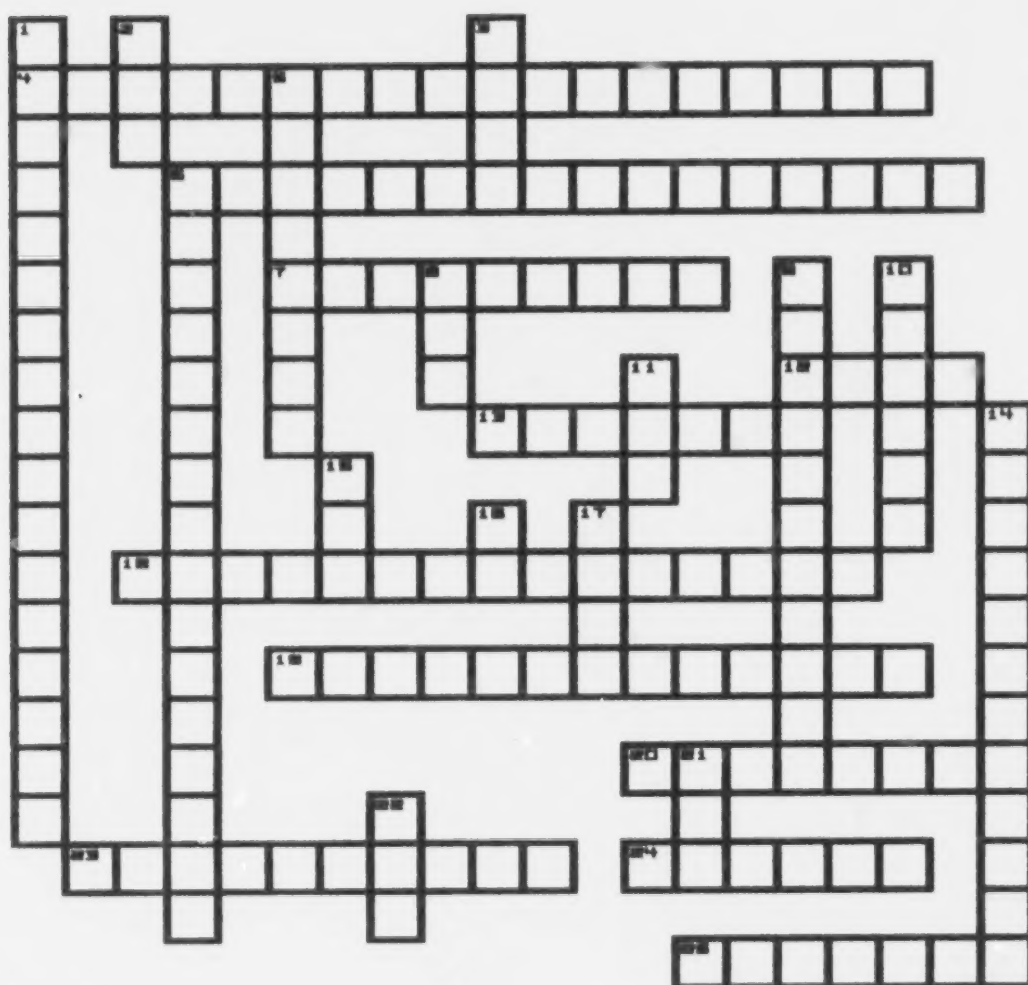
B. Disaster and recovery plans

1. Basic physical threats
  - a. Fire
  - b. Natural disaster
  - c. Environmental problems
  - d. Sabotage
2. Elements of a disaster recovery plan
  - a. Priorities
  - b. Personnel requirements
  - c. Equipment requirements

BCPI/Computer Hardware Utilization  
Unit 3: Computer Operation and Maintenance  
Student's Guide/UNIT OVERVIEW

- d. Facilities
    - e. Restore/recover utilities
- C. Backup and recovery
  - 1. **Backup procedures**--frequency, medium, storage, techniques, extent
  - 2. **Recovery procedures**--scheduled and unscheduled
- D. Copyright and ethics
  - 1. **Copyright laws**--scope and interpretation
    - a. Scope
    - b. Interpretation
    - c. Penalties
  - 2. Business ethics
    - a. Employee responsibility
    - b. Reporting infractions
    - c. Monetary implications
    - d. Inside information

### ACTIVITY 1: TERMINOLOGY CROSSWORD PUZZLE



## ACTIVITY 1: TERMINOLOGY CROSSWORD PUZZLE CLUES

### ACROSS CLUES

4. Programs that run on the computer to solve a particular application.
6. Individual responsible for setting up the computer system for execution.
7. Switch used in the case of an emergency to cancel computer's electricity supply.
12. Individual program incorporated into a job.
13. Data processing operation that is not under the direct control of the computer.
18. Standard that vendors informally accept because it has dominated the market.
19. Program supplied by the vendor as part of the operating system that combines the user's object modules with modules from the core image library.
20. Permits a program that requires more memory than is available to be executed.
23. Area within the CPU's memory.
24. A type of storage used to store programs or data that exist on an auxiliary device.
25. Method for inquiring whether a terminal has data to be transmitted.
3. Computer that provides the interface between terminals and the software supporting communication devices and terminals.
5. Program used to translate high-level language statements written in a programming language into machine language.
6. Portion of the SYSRES pack reserved for storing ready to execute programs such as compilers and application software.
8. Another term used to denote operating system software.
9. Program which checks each source code statement for errors and immediately translates the statement to object code and executes it.
10. Disk pack that must be on-line for the computer system to be operational.
11. Responsible for performing mathematical operations.
14. Operating systems that allow concurrent tasks.
15. Specialized language for setting up jobs and communicating basic instructions to the computer.
16. Addressable spaces on a secondary storage device that appear to the programmer as real memory.
17. An addressable portion of a program or data.
21. Operating system software with features normally not included in a typical VS system.
22. Procedure initiated by a control card or command entered through a terminal or console typewriter.

### DOWN CLUES

1. Group of computers that have the same general architecture.
2. Indicators and internal switches that are reset to load the supervisor into the CPU.

## ACTIVITY 2: OPERATE AND MAINTAIN MACHINES

**INSTRUCTIONS:** Visit facilities that would allow you to observe the following activities. Be prepared to write a brief summary of what you saw and to participate in a class discussion.

Observe:

Operator entering data via terminal/CRT and discuss typical messages and problems encountered. Summarize the discussion for class discussion.

A demonstration of microfiche use on a viewer and become familiar with methods for retrieval and display of information.

Observe how optical scanning devices are used and how unreadable and/or damaged documents/items are handled.

Visit a computer store and examine the difference between an external and internal direct-connect modem.



### ACTIVITY 3: MONITOR AND REPORT ENVIRONMENTAL PROBLEMS

#### Project 1

**INSTRUCTIONS:** Participate in a discussion of the effect of variations in weather upon various computer materials. Take notes.

Paper

Diskettes

Modems

CPU

Keyboard

#### Project 2

**INSTRUCTIONS:** Work in a group and find information in manufacturers' manuals for the school's computer and peripheral devices regarding the minimum and maximum recommended temperatures and humidity levels for operation and for storage.

### ACTIVITY 3 (Continued)

#### Project 3

**INSTRUCTIONS:** Complete the following exercise in conversion between Fahrenheit and Celsius temperature scales.

**Fahrenheit**

250

75

23

0

89

360

181

137

34

16

**Celsius**

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

4. \_\_\_\_\_

5. \_\_\_\_\_

6. \_\_\_\_\_

7. \_\_\_\_\_

8. \_\_\_\_\_

9. \_\_\_\_\_

10. \_\_\_\_\_

### ACTIVITY 3 (Continued)

#### Project 4

**INSTRUCTIONS:** Using the form provided, with the class, record the temperature and humidity of the computer room or other designated location at regular intervals for ten school days. Report any variations outside the acceptable range.

#### Project 5

**INSTRUCTIONS:** Using the form provided, record the temperature and humidity in your home or other selected location at regular intervals for ten days. Complete a report listing any variations outside the acceptable range.

#### Project 6

**INSTRUCTIONS:** For your assigned period of time, using the form provided, record the temperature and humidity of the computer room or other designated location at regular intervals for ten school days. Report any variations outside the acceptable range.

### ACTIVITY 3 (Continued)

Name \_\_\_\_\_

Place (circle one)    home            school            other \_\_\_\_\_

Specific location \_\_\_\_\_

| DATE  | TIME  | TEMPERATURE | HUMIDITY |
|-------|-------|-------------|----------|
| _____ | _____ | _____       | _____    |
| _____ | _____ | _____       | _____    |
| _____ | _____ | _____       | _____    |
| _____ | _____ | _____       | _____    |
| _____ | _____ | _____       | _____    |
| _____ | _____ | _____       | _____    |
| _____ | _____ | _____       | _____    |
| _____ | _____ | _____       | _____    |
| _____ | _____ | _____       | _____    |

Place an asterisk next to any variations outside the acceptable range. Discuss when relevant.

## ACTIVITY 4: INVENTORY RESOURCES AND REPORT FINDINGS

### Project 1

**INSTRUCTIONS:** List the preparation steps which involve the staging forms.

### Project 2

**INSTRUCTIONS:** Discuss reasons why a physical inventory might differ from the paper records of inventory items.

### Project 3

**INSTRUCTIONS:** Prepare a list of some of the criteria for determining where an item should be stored--size, shelf life, and frequency of access.

## ACTIVITY 4 (Continued)

### Project 4

**INSTRUCTIONS:** From a list of items prepared by your instructor, tell how each should be stored, including a reason for the choice.

### Project 5

**INSTRUCTIONS:** Using several catalogs from vendors and suppliers of computer merchandise, compile price and descriptive information for comparison purposes.

## ACTIVITY 5: HANDLE MEDIA AND SUPPORT TOOLS

### Project 1

**INSTRUCTIONS:** Discuss the reasons for off-site storage of backup copies of all valuable data disk and tapes.

### Project 2

**INSTRUCTIONS:** Prepare a summary chart about media life and recommended storage conditions after studying manufacturers' literature.

## ACTIVITY 5 (Continued)

### Project 3

**INSTRUCTIONS:** Describe five problems that may result from improper media handling.

### Project 4

**INSTRUCTIONS:** From your reading and research, describe the advancements made from the original quality of diskettes to the current Teflon-coated, high-quality disks.

### Project 5

**INSTRUCTIONS:** List the steps involved in the initialization process with and without operating system components. Initialize a diskette according to your instructor's instructions.



## ACTIVITY 5 (Continued)

### Project 6

**INSTRUCTIONS:** From your reading and research, describe the grandfather, father, and son methods of updating files, and explain the rationale.

### Project 7

**INSTRUCTIONS:** Prepare a list of the items of information that are needed for properly labeling diskettes.

## ACTIVITY 6: MONITOR HARDWARE PERFORMANCE

**INSTRUCTIONS:** - With an error message section from a manual, look up the common diagnostic messages provided by your instructor to determine the causes of problems and determine the options available for recovery.

## ACTIVITY 7: SCHEDULE WORKFLOW

### Project 1

**INSTRUCTIONS:** From the run instructions for sample jobs provided by your instructor, determine:

1. Resources needed (paper, magnetic media, etc.)
2. Distribution
3. Special instructions for exception reporting, restart, and programmed messages
4. Data controls

### Project 2

**INSTRUCTIONS:** Given a written set of specifications, complete a run instruction sheet for sample jobs provided by your instructor.

## ACTIVITY 8: MAINTAIN DATA AND PHYSICAL SECURITY

### Project 1

**INSTRUCTIONS:** From an applications program provided by your instructor, try to access files. Describe what happens when unauthorized users are denied access to the system or limited to certain data.

### Project 2

**INSTRUCTIONS:** Read several current articles about items that should be included in a disaster recovery plan. Write a report on one.

## **SECTION III**

### **PROGRAM DEVELOPMENT**

#### **ESSENTIAL ELEMENT**

**75.87(g)(3)(4)(5)**

3. Concepts and skills associated with algorithmic approaches to problem solving. The student shall be provided opportunities to:
  - (A) Demonstrate the skills prelated to structured approaches to problem solving; and
  - (B) Translate problems into logical steps using appropriate program design techniques.
  
4. Concepts and skills associated with commands and statements of a programming language. The student shall be provided opportunities to:
  - (A) Identify and use variables, constants, and literals;
  - (B) Demonstrate the use of input and output statements;
  - (C) Demonstrate and use conditional and unconditional branching statements;
  - (D) Demonstrate and use of looping techniques; and
  - (E) Develop and use subroutines for modular programming.
  
5. Concepts and skills related to program execution, testing, debugging, and documentation. The student shall be provided opportunities to:
  - (A) Demonstrate the ability to execute computer programs with sample data;
  - (B) Distinguish between syntax and logic errors;
  - (C) Apply standard debugging procedures; and
  - (D) Develop internal and external program documentation including record layout and report spacing chart.

## SUGGESTED UNIT LESSON PLAN

### SECTION III: PROGRAM DEVELOPMENT

#### UNIT I: PROBLEM-SOLVING CONCEPTS AND APPLICATIONS

|                               |                               |                                     |
|-------------------------------|-------------------------------|-------------------------------------|
| <b>ESSENTIAL<br/>ELEMENT:</b> | <b>75.87<br/>(g)(3)(A)(B)</b> | <b>ALLOCATED<br/>TIME:</b> 10 Hours |
|-------------------------------|-------------------------------|-------------------------------------|

**UNIT  
OBJECTIVE:** Students will demonstrate the skills related to algorithmic approaches to problem solving and translate problems into logical steps using appropriate program design techniques.

**STUDENT  
COMPETENCIES:** After completing this unit, students will be able to:

1. Define structured programming and related terminology.
2. Identify three types of control structures.
3. Demonstrate problem-solving techniques.
4. Utilize flowcharting and top-down analysis.

**FOCUS:** In order to be able to write structured programs in an acceptable time frame, students must be skilled in the development of problem-solving techniques, program design, and program documentation.

**DOMAINS:** Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:** Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. Structured programming
- II. Control structures
  - A. The importance of
  - B. The three basic types
- III. Solving the problem
  - A. Problem definition
  - B. Information desired
  - C. Data required
  - D. Processing needed
- IV. Designing the solution
  - A. Top-down design
  - B. IPO chart
  - C. Algorithm
  - D. Flowchart
    - 1. Types
    - 2. Symbols
    - 3. Advantages
    - 4. Disadvantages
  - E. Pseudocode

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.



**Specific Teaching Activities**

1. Prepare a bulletin board from the following ideas:

Computer languages  
Flowcharting/Pseudocode  
Program development cycle  
Debugging

2. Have available:

Student formatted data disk system master with BASIC flowcharting templates.

Flowcharting templates.

Textbooks or other materials that explain various high-level languages and applications to which students are best-suited.

Examples of top-down design, IPO charts, pseudocode, and flowcharts.

Copies of computer programs.

**Specific Reteach Activities**

1. Have students write instructions for making a peanut butter and jelly sandwich. The next day bring the ingredients and materials needed to make a peanut butter and jelly sandwich. Have students attempt to make the sandwich from the instructions they have written.
2. Have students write step-by-step instructions for calculating the estimated cost of carpeting a room. This would involve a figure for width, length, and the cost per yard of carpet. Calculations need to be included for square feet, square yards, and total cost. Have the students write each step on a separate index card. Students should mix up the cards and have another student place them in the proper order. This activity could also be done with calculating net pay, a water bill, or the total bill of a purchase of three parts from an auto supply store.
3. Use two simple programs, perhaps some type of demo disk or a drill program. The students should write documentation for one of the programs. The students should then take the documentation written for the other program and attempt to run the program following the instructions in the documentation.

**BCPI/Program Development**  
**Unit 1: Problem Solving Concepts and Applications**  
**Teacher's Guide/ACTIVITY MATRIX**

### ACTIVITY MATRIX

[illegible]

**BCPI/Program Development**  
**Unit 1: Problem Solving Concepts and Applications**  
**Teacher's Guide/COMPETENCY TEST MATRIX**

## COMPETENCY TEST MATRIX

[illegible]

## HANDOUT

### EXAMPLE OF AN IPO CHART

**PROBLEM:** Write a problem solution to find the selling price of given items. The input contains the item description and cost. The selling price is the cost of the item plus 45% markup. On the report print the item description, the cost, the markup, and the selling price.

| I -- INPUT                                                                               | P -- PROCESSING                                                                              | O -- OUTPUT                                               |
|------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|-----------------------------------------------------------|
| ** In this column put everything you know about the information you are using for input. | ** In this column put the calculations you are going to have to make to get an output field. | ** In this column put fields that you want on the report. |

ITEM DESCRIPTION  
COST

MARKUP = COST \* .45  
SELLING PRICE = COST + MARKUP

ITEM DESCRIPTION  
COST  
MARKUP  
SELLING PRICE

ANOTHER POSSIBILITY:

RATE = .45

MARKUP = COST \* RATE

MARKUP

## COMPETENCY TEST

### PART I: STRUCTURED PROGRAMMING AND TERMINOLOGY

#### Project 1

INSTRUCTIONS: Define structured programming.

#### Project 2

INSTRUCTIONS: Define the terms below.

1. Documentation
2. Machine language
3. Read and data
4. High-level languages
5. Controlled loops

## PART II: CONTROL STRUCTURES

INSTRUCTIONS: Describe the three control structures.

1.

2.

3.

### PART III: PROBLEM SOLVING

#### Project 1

**INSTRUCTIONS:** List the six steps in the Program Development Cycle.

- 1.
- 2.
- 3.
- 4.
- 5.
- 6.

#### Project 2

**INSTRUCTIONS:** Examine the following situation. Determine the problem, what information is desired, what data are required, and what processing will be needed.

Ruth Waller, senior English teacher, does not send out progress reports on time and has difficulty getting grades ready on time for report cards.

Identify the problem:

Information desired:

Data required:

Processing needed:

## PART IV: TOP-DOWN ANALYSIS AND FLOWCHARTS

### Project 1

INSTRUCTIONS: Explain top-down design.

### Project 2

INSTRUCTIONS: Develop pseudocode to calculate the cost of carpet for a room.



## PART IV (Continued)

### Project 3

**INSTRUCTIONS:** Draw a flowchart for determining the batting average of a baseball player.

### Project 4

**INSTRUCTIONS:** Explain the relationships between structured programming and top-down design.

## PART IV (Continued)

### Project 5

**INSTRUCTIONS:** Of the various ways of organizing programmers, explain which you would prefer.

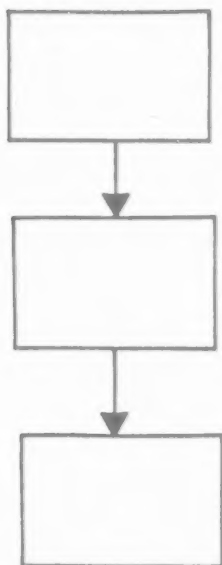
377

## ANSWERS TO ACTIVITIES

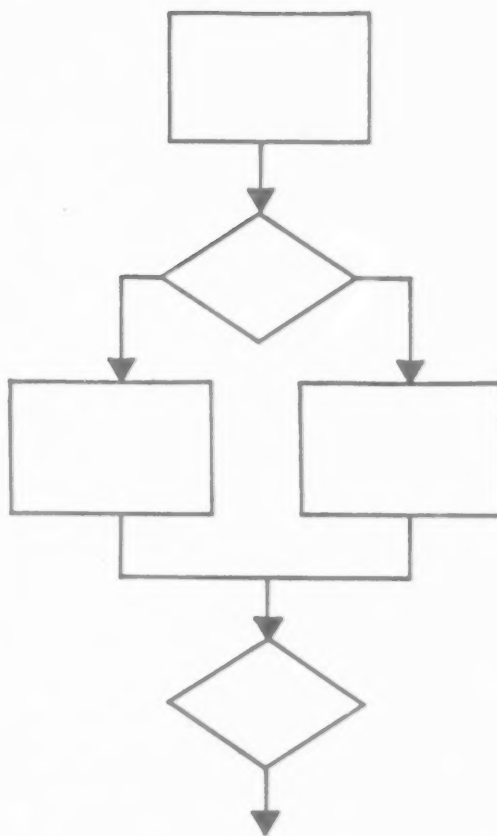
Activity 1     Evaluated to the satisfaction of the instructor.

Activity 2

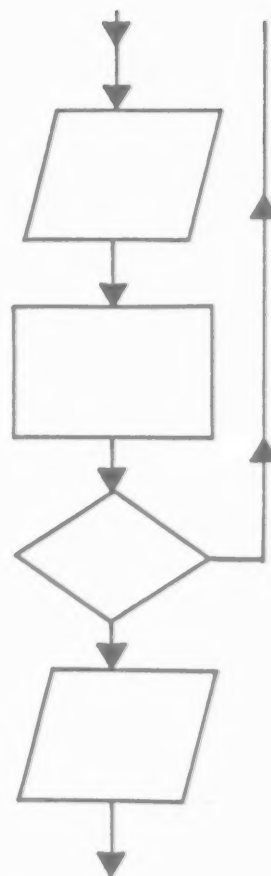
SEQUENCE



SELECTION



LOOP



### Activity 3

#### Project 1

Lowlands water supply seems to have a cash flow problem. It seems that their own bills are due each month before they receive payments for their customers.

- a. Money is not received soon enough from the customers. The basic problem might be the billing process. Lowlands' billing system takes too long. Perhaps they need a computerized billing system.
- b. A computer generated bill is needed. The bill should have the customer's name and address, the current meter reading, the previous meter reading, the gallons used, and the amount of the bill.
- c. In order to print the customer's bill properly, the system will need the billing address, last month's meter reading, the current month's meter reading, and the rate per 100 gallons the customer is charged.
- d. In order to calculate the bill, the program will have to subtract the previous reading from the current reading to determine gallons used. The program must then divide the gallons by 100 and multiply the rate per 100 gallons.

#### Project 2

Arnold's Electrical Supply never seems to have the right item when the customer comes in. On other items there seems to be an oversupply.

- a. There is poor information concerning the amount of inventory. There needs to be a more accurate picture of the inventory.
- b. There needs to be an accurate list of the inventory on hand. Information needed should be the part number and the amount currently in stock.
- c. The business must have the part number of all parts, the amount of beginning inventory, and the number of items sold.
- d. The program must be able to take the beginning inventory and subtract the parts as they are sold to show current inventory levels.

### Activity 4

#### Project 1

- 1.-2. Evaluated to the satisfaction of the instructor.

Activity 4 (Continued)

Project 2

1. Sum the major grades.  
Divide the sum of major grades by number of grades.  
Sum daily grade.  
Divide sum of daily grades by number of grades.  
Grade for six weeks' test.  
Sum: Major average times two plus daily average plus six weeks' test grade.  
Take sum of averages and six weeks' test and divide by four.
2. Sum number of hours worked.  
Gross pay: Multiply number of hours times rate of pay.  
Calculate withholding.  
Calculate FICA.  
Determine amount of insurance.  
Net Pay = Gross Pay - (withholding - FICA-Insurance).
3. Determine price of each part.  
Multiply number of parts times price for each part.  
Sum cost of parts.  
Multiply tax rate times sum of parts.  
Add tax to sum of parts.  
Determine labor rate per hour.  
Labor expense: Multiply number of hours times labor rate per hour.  
Total Bill: Sum parts, tax on parts, and labor expense.

Project 3

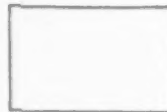
1. Start or End



2. Input or Output



3. Processing



4. Decision



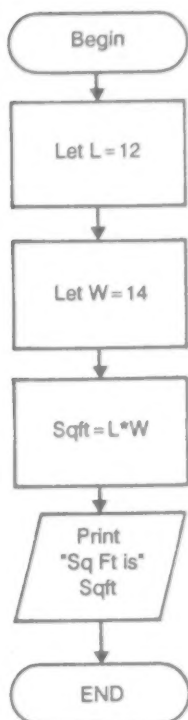
5. Connection between parts of a flowchart



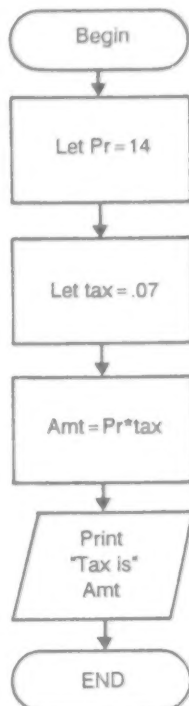
# Activity 4 (Continued)

## Project 4

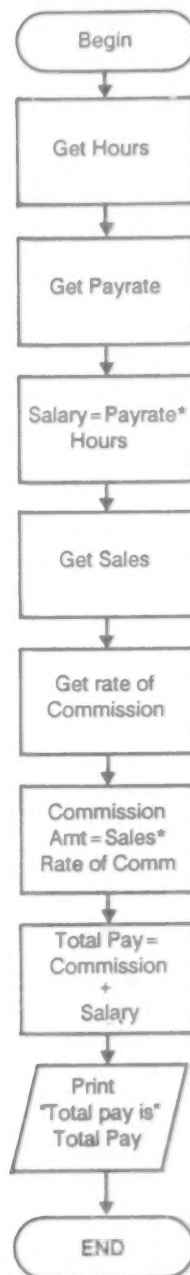
1.



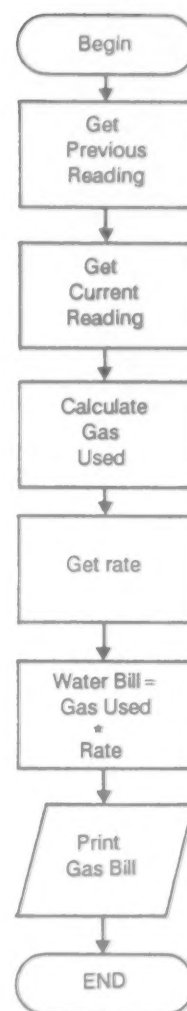
2.



3.



4.



#### Activity 4 (Continued)

##### Project 5

1. Determine beginning inventory.  
Determine number of items sold.  
 $\text{Ending inventory} = \text{beginning inventory} - \text{number sold}.$
2. Determine cost of item.  
Determine sale price of item.  
 $\text{Profit each} = \text{sale price} - \text{cost of item}.$   
 $\text{Gross Profit} = \text{profit each} \times \text{number sold}.$
3. Determine cost per night of room.  
Multiply cost per night times 2.  
Determine phone charges, taxes and any other charges.  
Sum all costs.  
Divide the sum by 4.

##### Project 6

1. Typical data--grades between 0 and 100. Unusual, but valid data--grades between 20 and 120. Incorrect or inappropriate data--grades of 200 or 1200 or other extremes. Entering words when numbers are requested.
2. Typical data--enter part numbers in stock, a normal quantity. Check to see if the part is taken out of stock.

Unusual, but valid data--enter a seldom requested part number. Enter an unusually large quantity on several items. Enter a large number of items to be purchased. Enter a quantity larger than the number in stock.

Incorrect or inappropriate data--enter part numbers incorrectly (use words instead of numbers, wrong amount of numbers in a part number). Enter an inappropriate quantity (ex. 1,000,000). Enter a negative number for quantity.

## COMPETENCY TEST ANSWERS

### Part I

#### Project 1

Structured programming is a logical style of design at the beginning of project. A structured program is a program in which each block of code (or module) is made up of one of three logical control structures: sequence, selection, or looping.

#### Project 2

**Documentation** is the written explanation of the program, which is important to the programmer as well as to the program user. Good documentation is important if there is ever a need to modify or improve a program.

**Machine language** is inside the computer, instructions are processed electronically in the binary language of the computer, machine language. Machine code is very time-consuming and difficult to write.

**Read and data** permit information to be stored in the program. It is a method of assigning a value to a variable.

**High-level languages** are languages that use English-like terms and commands to make programming and understanding programs written by others easier. In most applications they are easier to learn, require less time to write, are not machine dependent, and are easier to understand.

**Controlled loops** are used to repeat an action. Controlled loops enable the programmer to cause an action to be repeated a certain number of times or until a certain condition exists.

### Part II

1. Sequence is a structure pattern used to execute statements one after another in order of storage. After all statements have been executed, the problem is solved.
2. Selection structure is used to compare data in a program. This pattern requires a test.
  - a. The comparison determines the action the program will take.
  - b. The choices or decisions split a program to follow different paths.
  - c. A logical test usually is the reason for this structure.
    - (1) If the test is proven to be true, one path of the program and logic flow is taken by the computer.
    - (2) If the test proves false, then another path is taken.



## Part II (Continued)

3. Loop structure tests the number of times the loop is executed.
  - a. Each time the loop is executed, the pass through is called an iteration.
  - b. There are two major types of loops.
    - (1) A DOWHILE loop continues to occur as long as the condition is TRUE. The test occurs immediately upon ENTERING the loop.
    - (2) A DOUNTIL loop test is at the END of the loop after all other processing within the loop has occurred. The basis of this loop is that the given condition is FALSE, and the loop continues to execute as long as that condition remains false.

## Part III

### Project 1

1. Problem solving
2. Designing the solution
3. Coding the program
4. Debugging
5. Testing
6. Documentation

### Project 2

The teacher does not have the time to average all the students' grades and type progress reports.

Student name, list of grades, and average. At the end of each third week, a progress report is needed.

Student names and grades.

The program will need to take the grades and produce an average. The program will need to determine if an average is below a certain number and then produce a progress report.

## Part IV

### Project 1

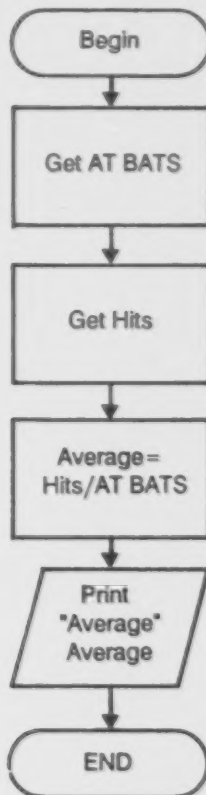
Top-down design involves taking a large problem and breaking it down into a series of smaller parts. Each smaller part is then broken down into a series of steps.

Part IV (Continued)

Project 2

Determine the length of room.  
Determine width of room.  
Square feet = length x width.  
Square yards = square feet/9.  
Determine cost per square yard of carpet.

Project 3



Part IV (Continued)

Project 4

Structured programming is a style of clear, organized programming. Top-down design takes a problem and breaks it down into small manageable parts. Problems broken down in this way are clearer, more organized, and easier to understand.

Project 5

Students should identify one type of organization (hierarchical grouping, chief programmer, "egoless" team) and give an explanation of why they would prefer that style.

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(3)(A)(B)

The students will be provided opportunities to: demonstrate the skills related to algorithmic approaches to problem solving and translate problems into logical steps using appropriate program design techniques.

---

---

**OBJECTIVE:** Students will demonstrate the skills related to algorithmic approaches to problem solving, and translate problems into logical steps using appropriate program design techniques.

|  | MASTERY | NON-MASTERY |
|--|---------|-------------|
|--|---------|-------------|

---

---

|                                                           |       |       |
|-----------------------------------------------------------|-------|-------|
| 1. Define structured programming and related terminology. | _____ | _____ |
|-----------------------------------------------------------|-------|-------|

|                                                |       |       |
|------------------------------------------------|-------|-------|
| 2. Identify three types of control structures. | _____ | _____ |
|------------------------------------------------|-------|-------|

|                                            |       |       |
|--------------------------------------------|-------|-------|
| 3. Demonstrate problem-solving techniques. | _____ | _____ |
|--------------------------------------------|-------|-------|

|                                                |       |       |
|------------------------------------------------|-------|-------|
| 4. Utilize flowcharting and top-down analysis. | _____ | _____ |
|                                                | _____ | _____ |
|                                                | _____ | _____ |
|                                                | _____ | _____ |

STUDENT NAME \_\_\_\_\_ 387 \_\_\_\_\_

## SUGGESTED LESSON PLAN

### SECTION III: PROGRAM DEVELOPMENT

#### UNIT II: PROGRAMMING CONCEPTS AND APPLICATIONS

|                               |                                                                   |                                     |
|-------------------------------|-------------------------------------------------------------------|-------------------------------------|
| <b>ESSENTIAL<br/>ELEMENTS</b> | <b>75.87</b><br>(g)(3)(A)(B)(4)(A)(B)(C)(D)(E)<br>(5)(A)(B)(C)(D) | <b>ALLOCATED<br/>TIME: 55 Hours</b> |
|-------------------------------|-------------------------------------------------------------------|-------------------------------------|

**UNIT  
OBJECTIVE:** Students will demonstrate the skills associated with a programming language and the skills related to program execution, testing, debugging, and documentation.

**STUDENT  
OBJECTIVES:** After completing these modules, the students will be able to:

#### MODULE A: SETUP PROCEDURES

1. Demonstrate the ability to power-up, load the compiler, and power-down the computer.
2. List the steps of problem solving.

#### MODULE B: INTRODUCTION TO PROGRAMMING

1. Identify keywords and commands.
2. Code, key, save, load, and execute a program.
3. Edit program lines.
4. Write simple programs using the keywords REM, PRINT,END.

#### MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING

1. Explain the benefits of documentation.

2. Identify the three major types of errors.
3. Develop test data for a program.

#### **MODULE D: USING ARITHMETIC**

1. Define the types of variables.
2. Use arithmetic operators.
3. Develop arithmetic equations.
4. Write programs using the keyword LET.

#### **MODULE E: INPUT CONCEPTS**

1. Demonstrate the use of data terminators.
2. Write programs using the keywords READ/DATA, IF...THEN, GOTO, and INPUT.
3. Demonstrate an output report by using proper headings (major, minor, and column), and PRINT USING.

#### **MODULE F: BRANCHING**

1. Identify the relational operators.
2. Identify the logical operators.
3. Describe the difference between a conditional branch and an unconditional branch.
4. Write programs using the conditional branch.

#### **MODULE G: TOTALS**

1. Define totaling and counting.

2. Demonstrate how subtotals are accumulated and printed.
3. Write programs to demonstrate totaling, counting, and subtotals.

#### MODULE H: MODULE PROGRAMMING

1. Define modular programming.
2. Write programs to use a main module and submodules.

#### MODULE I: CONTROLLED LOOPS

Write programs using the keywords FOR/NEXT.

#### MODULE J: ADVANCED CONCEPTS

1. Write programs using tables to store data.
2. Write programs using sort routines.
3. Write programs using advanced character handling.
4. Write programs using graphics.

**FOCUS:** Each programming language has its own set of rules, structure, and syntax. When students understand the structure and syntax of a language and develop program design skills, they are able to use a high-level language to solve problems and program a computer.

**DOMAINS:** Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:** Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.



## CONTENT OUTLINE

### MODULE A: SETUP PROCEDURES

- I. Powering-up, loading the compiler, and powering-down
- II. Studying programming
  - A. Concepts for successful problem solving
  - B. More than one correct way
  - C. Programming assignments
  - D. Reading and studying the material
  - E. Programming process

### MODULE B: INTRODUCTION TO PROGRAMMING

- I. Using a high-level language
  - A. Semantics and syntax
  - B. Using a high-level language for activities
  - C. Keywords and commands
- II. Coding the program
  - A. Programmer organization
  - B. Structure
- III. Editing
  - A. Adding lines
  - B. Deleting lines
  - C. Modifying lines

### MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING

- I. Documentation
  - A. Benefits
  - B. Internal
  - C. External
  - D. Company guidelines
- II. Debugging
  - A. Errors
  - B. Common errors

- C. Debugging aids
- III. Testing
  - A. Purpose
  - B. Procedure
  - C. Test data creation
  - D. Interpretation

#### MODULE D: USING ARITHMETIC

- I. Variables
  - A. Rules
  - B. Types
- II. Arithmetic
  - A. Hierarchy of operations
  - B. Symbols
- III. Using the keyword LET
  - A. Value content
  - B. Receiving variable position
  - C. Optional feature

#### MODULE E: INPUT CONCEPTS

- I. Data terminators
- II. Input statements
  - A. READ and DATA
  - B. INPUT
  - C. GOTO
  - D. IF...THEN
- III. Output

#### MODULE F: BRANCHING

- I. Relational operators
  - A. Equal to
  - B. Less than
  - C. Greater than

- D. Not equal to
- E. Less than or equal to
- F. Greater than or equal to
- II. Logical operators
  - A. AND
  - B. OR
  - C. NOT
- III. Branching
  - A. GOTO
  - B. IF...THEN
  - C. Looping
  - D. Data validation

#### MODULE G: TOTALS

- I. Totaling and counting
  - A. Totals
  - B. Counters
  - C. Conditional and unconditional accumulators
- II. Subtotals
  - A. Control variables
  - B. Compare variables
  - C. Subtotal process

#### MODULE H: MODULE PROGRAMMING

- I. Subroutines
  - A. One for each function
  - B. Main module
  - C. Planning submodules
- II. GOSUB...RETURN
  - A. GOSUB
  - B. RETURN

## MODULE I: CONTROLLED LOOPS

- I. Looping
  - A. Using the keywords FOR/NEXT
  - B. Format for the FOR/NEXT statement
  - C. Constants or variables
  - D. STEP option
  - E. FOR/NEXT examples
  - F. Nested loops

## MODULE J: ADVANCED CONCEPTS

- I. Tables
  - A. Elements
  - B. Subscripts
  - C. DIM
  - D. READ or INPUT
  - E. Searching
- II. Sorting
  - A. Comparing
  - B. Swapping elements
  - C. Repetition
  - D. Several methods
- III. Various keywords
  - A. LEFT\$
  - B. RIGHT\$
  - C. MID\$

BCPI/Program Development  
Unit 2: Programming Concepts and Applications  
Teacher's Guide/CONTENT OUTLINE

- D. LEN
- E. INKEY\$
- F. STR\$
- G. VAL

IV. Graphics

- A. SCREEN
- B. COLOR
- C. PSET
- D. LINE
- E. CIRCLE
- F. DRAW
- G. PAINT

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

### Specific Teaching Activities

1. A textbook or a combination of textbooks will be necessary for reference, more programs, and other activities.
2. Prepare a bulletin board from the following ideas:

Computer languages  
BASIC commands and keywords  
Flowcharting/Pseudocode  
Debugging  
Documentation  
Coding of BASIC programs

3. Have available:

Student formatted data disks, system master with language  
Flowcharting templates  
Examples of computer programs

4. FOR EACH PROGRAMMING MODULE:

#### STUDENT ACTIVITY DEBUGGING--SYNTAX ERRORS

Have the students put 3-5 syntax errors in previous programs they have completed. Compile these programs, and get two source listings and output if possible. Let the students mark the errors on one copy. Then the students should hand both copies into the instructor. Use the unmarked copy for a debugging exercise, quiz or test. The key is already made.

#### STUDENT ACTIVITY DEBUGGING--LOGIC ERRORS

Have the students put logic errors in previous programs they have completed. Compile these programs, and get two source listings and the output that is possible. Let the students mark the errors on one copy. Then the student should hand both copies into the instructor. Use the unmarked copy for a debugging exercise, quiz or test. The key is already made.

5. Have students work with other students as peer tutors.
6. Some activities and some parts of the competency test have been combined. You may wish to separate these activities and parts or add to them.

### MODULE C

Activity 1, Project 1 will need two programs.  
Activity 1, Project 2 will need written documentation.

Specific Reteach Activities

1. Show a film or filmstrip on high-level languages and the applications to which they are best suited.
2. Have students load programs previously written by other students or the instructor.
3. Have students load programs previously written by other students and modify the programs.
4. Have students work in pairs to type in programs from a textbook, save the programs, load the programs, modify the programs, and save the modified programs with different names.
5. Use an Algebra I book to show how algebra is done using variables.
6. Have students view hard copies of successful programs of other students. Have students write the purpose of each line and flowchart the program.
7. Have a student who has successfully written a program write his program on the overhead or board. The student should then explain each step of the program to the class.
8. Have students work in groups to write a program. The group leaders should assign the parts of the program to various group members. The parts can be merged.
9. Have students work problems on paper, divide the problems into steps, write a line of code for each step, and then key the program into the computer.
10. You may wish to explore other programming features such as the ELSE statement and how the colon is used within a program.



**BCPI/Program Development**  
**Unit 2: Programming Concepts and Applications**  
**Teacher's Guide/ACTIVITY MATRIX**

**ACTIVITY MATRIX**

| Activities  | Domains   |              |           | Basic Skills |         |      |
|-------------|-----------|--------------|-----------|--------------|---------|------|
|             | Cognitive | Psycho-motor | Affective | Reading      | Writing | Math |
| MODULE A    |           |              |           |              |         |      |
| 1 Project 1 | X         | X            |           |              |         |      |
| Project 2   | X         | X            |           |              | X       | X    |
| 2 Program 1 | X         | X            |           |              | X       |      |
| 2           | X         | X            |           |              | X       |      |
| 3           | X         | X            |           |              | X       |      |
| MODULE B    |           |              |           |              |         |      |
| 1           | X         |              |           | X            |         |      |
| 2           | X         | X            |           |              | X       |      |
| 3 Project 1 | X         | X            |           |              | X       |      |
| Project 2   | X         | X            |           |              | X       |      |
| Project 3   | X         | X            |           |              | X       |      |
| 4 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         | X            |           |              | X       |      |
| Project 3   | X         | X            |           |              | X       |      |
| Project 4   | X         | X            |           |              | X       |      |
| Project 5   | X         | X            |           |              | X       |      |
| MODULE C    |           |              |           |              |         |      |
| 1 Project 1 | X         |              |           | X            | X       |      |
| Project 2   | X         |              |           | X            | X       |      |
| 2           | X         |              |           |              | X       |      |
| 3 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         |              |           |              | X       |      |
| MODULE D    |           |              |           |              |         |      |
| 1           | X         |              |           | X            |         |      |

## ACTIVITY MATRIX

| Activities       | Domains   |              |           | Basic Skills |         |      |
|------------------|-----------|--------------|-----------|--------------|---------|------|
|                  | Cognitive | Psycho-motor | Affective | Reading      | Writing | Math |
| MODULE D (Con't) |           |              |           |              |         |      |
| 2 Project 1      | X         | X            |           |              | X       | X    |
| Project 2        | X         | X            |           |              | X       | X    |
| Project 3        | X         | X            |           |              | X       | X    |
| 3                | X         |              |           |              |         | X    |
| 4 Project 1      | X         |              |           |              | X       |      |
| Project 2        | X         | X            |           |              | X       | X    |
| MODULE E         |           |              |           |              |         |      |
| 1-2 Project 1    | X         |              |           |              | X       | X    |
| Project 2        | X         |              |           |              | X       |      |
| Project 3        | X         |              |           |              | X       |      |
| Project 4        | X         |              |           |              | X       |      |
| Project 5        | X         |              |           |              | X       |      |
| Project 6        | X         |              |           |              | X       |      |
| 3 Project 1      | X         | X            |           |              | X       |      |
| Project 2        | X         | X            |           |              | X       |      |
| Project 3        | X         |              |           |              | X       |      |
| Project 4        | X         |              |           |              | X       |      |
| Program 1        | X         | X            |           |              | X       |      |
| 2                | X         | X            |           |              | X       | X    |
| 3                | X         | X            |           |              | X       | X    |
| 4                | X         | X            |           |              | X       |      |
|                  |           |              |           |              |         |      |
|                  |           |              |           |              |         |      |
|                  |           |              |           |              |         |      |

### ACTIVITY MATRIX

| Activities  | Domains   |              |           | Basic Skills |         |      |
|-------------|-----------|--------------|-----------|--------------|---------|------|
|             | Cognitive | Psycho-motor | Affective | Reading      | Writing | Math |
| MODULE F    |           |              |           |              |         |      |
| 1 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         |              |           |              | X       |      |
| 2           | X         |              |           |              | X       |      |
| 3 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         |              |           |              | X       |      |
| Program 1   | X         | X            |           |              | X       |      |
| 2           | X         | X            |           |              | X       | X    |
| 3           | X         | X            |           |              | X       | X    |
| MODULE G    |           |              |           |              |         |      |
| 1           | X         |              |           |              | X       |      |
| Program 1   | X         | X            |           |              | X       | X    |
| 2           | X         | X            |           |              | X       | X    |
| 3           | X         | X            |           |              | X       | X    |
| MODULE H    |           |              |           |              |         |      |
| Program 1   | X         | X            |           |              | X       | X    |
| 2           | X         | X            |           |              | X       | X    |
| MODULE I    |           |              |           |              |         |      |
| 1 Project 1 | X         |              |           |              | X       |      |
| Project 2   | X         |              |           |              | X       |      |
| Program     | X         | X            |           |              | X       | X    |
| MODULE J    |           |              |           |              |         |      |
| 1           | X         |              |           |              | X       |      |
| Program     | X         | X            |           |              | X       |      |
|             |           |              |           |              |         |      |

### COMPETENCY MATRIX

| Parts           | Domains   |              |           | Basic Skills |         |      |
|-----------------|-----------|--------------|-----------|--------------|---------|------|
|                 | Cognitive | Psycho-motor | Affective | Reading      | Writing | Math |
| MODULE A        |           |              |           |              |         |      |
| I               | X         |              |           |              | X       |      |
| II              | X         |              |           |              | X       |      |
| MODULE B        |           |              |           |              |         |      |
| I Project 1     | X         |              |           | X            |         |      |
| Project 2       | X         |              |           |              | X       |      |
| II-IV Project 1 | X         |              |           |              | X       |      |
| Project 2       | X         | X            |           |              | X       |      |
| Project 3       | X         | X            |           |              | X       |      |
| Project 4       | X         | X            |           |              | X       |      |
| Project 5       | X         | X            |           |              | X       |      |
| MODULE C        |           |              |           |              |         |      |
| I Project 1     | X         |              |           |              | X       |      |
| Project 2       | X         |              |           |              | X       |      |
| II              | X         |              |           |              | X       |      |
| III             | X         |              |           |              | X       |      |
| MODULE D        |           |              |           |              |         |      |
| I Project 1     | X         |              |           |              | X       |      |
| Project 2       | X         |              |           |              | X       |      |
| II Project 1    | X         |              |           |              | X       |      |
| Project 2       | X         |              |           |              | X       |      |
| III             | X         |              |           |              |         | X    |
| IV              | X         |              |           |              | X       |      |
|                 |           |              |           |              |         |      |
|                 |           |              |           |              |         |      |

### COMPETENCY TEST MATRIX

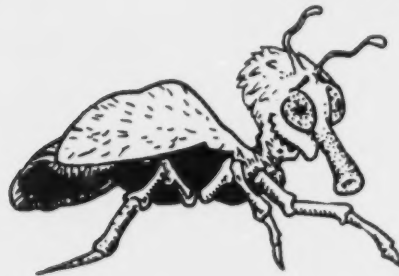
| Parts            | Domains   |              |           | Basic Skills |         |      |
|------------------|-----------|--------------|-----------|--------------|---------|------|
|                  | Cognitive | Psycho-motor | Affective | Reading      | Writing | Math |
| MODULE E         |           |              |           |              |         |      |
| I & II Project 1 | X         |              |           |              | X       |      |
| Project 2        | X         |              |           |              | X       |      |
| Project 3        | X         |              |           |              | X       |      |
| III Project 1    | X         | X            |           |              | X       |      |
| Project 2        | X         | X            |           |              | X       |      |
| MODULE F         |           |              |           |              |         |      |
| I                | X         |              |           |              | X       |      |
| II               | X         |              |           |              | X       |      |
| III              | X         |              |           |              | X       |      |
| IV               | X         | X            |           |              | X       |      |
| MODULE G         |           |              |           |              |         |      |
| I                | X         |              |           |              | X       |      |
| II & III         | X         | X            |           |              | X       | X    |
| MODULE H         |           |              |           |              |         |      |
| I                | X         |              |           |              | X       |      |
| II Project 1     | X         |              |           |              | X       |      |
| Project 2        | X         |              |           |              | X       |      |
| MODULE I         |           |              |           |              |         |      |
| Project 1        | X         | X            |           |              | X       |      |
| Project 2        | X         | X            |           |              | X       |      |
| MODULE J         |           |              |           |              |         |      |
| I - IV           | X         | X            |           |              | X       |      |
|                  |           |              |           |              |         |      |
|                  |           |              |           |              |         |      |

## Debugging a Program

**“Bugs” are program errors - two kinds:**

1. **SYNTAX** - error in form of program language

To Eliminate:   a. Interpret diagnostics  
                    b. Desk check program code



2. **LOGIC** - error in flowchart or incorrect code statement

To Eliminate:   a. Interpret diagnostics  
                    b. Desk check flowchart  
                    c. Desk check code statements



IF...THEN RELATIONAL SYMBOLS

SYMBOL

MEANS

=

is equal to

>

is greater than

<

is less than

>=

is greater than or equal to

<=

is less than or equal to

<>

is not equal to

### LOGIC DECISIONS

| Storage |     |
|---------|-----|
| A       | B   |
| 500     | 500 |

1. Are A and B equal?
2. Are A and B not equal?
3. Is A less than B?
4. Is A greater than B?
5. Is A less than or equal to B?
6. Is A greater than or equal to B?

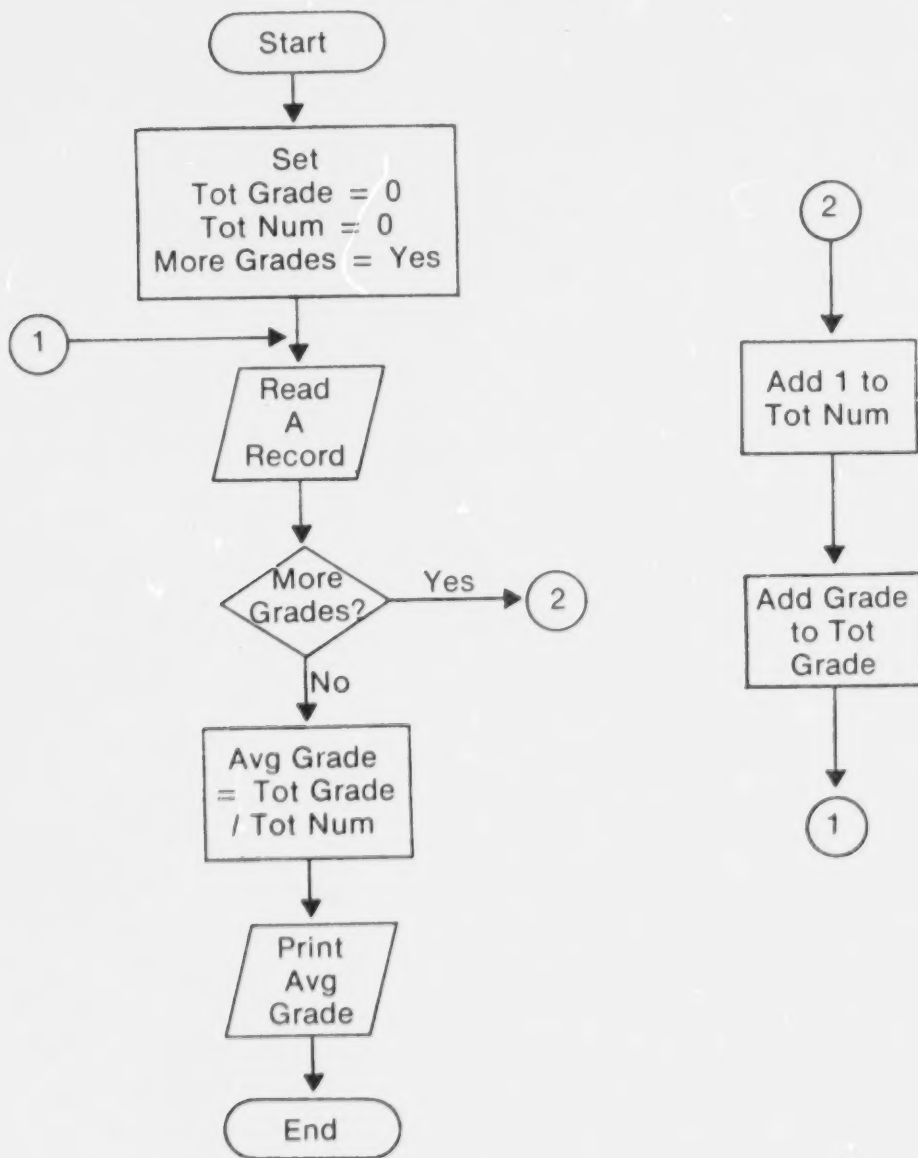


**PSEUDOCODE  
(PROGRAM DESIGN LANGUAGE)**

- Uses English language statements
- Used for program planning
- Easily written; modified
- Logic Structure

1. Sequence logic - instructions written in performance sequence
2. Decision logic - IFTHEN and IFELSE statements
3. Loop logic - DOWHILE and DOUNTIL statements

**PROGRAM FLOWCHART FOR PSEUDOCODE PROBLEM:  
AVERAGE GRADE**



## COMPETENCY TEST

### MODULE A: SETUP PROCEDURES

#### PART I: POWER-UP, LOADING, AND POWER-DOWN

**INSTRUCTIONS:** Write the instructions on how to power-up, load and power-down the computer.

#### PART II: PROBLEM SOLVING

**INSTRUCTIONS:** Write a paragraph discussing the programming process.

## COMPETENCY TEST

### MODULE B: INTRODUCTION TO PROGRAMMING

#### PART I: KEYWORDS AND COMMANDS

##### Project 1

**INSTRUCTIONS:** Matching the following terms with the correct keyword. Put the correct letter in the answer blank.

- |    |       |    |      |
|----|-------|----|------|
| A. | NEW   | D. | LOAD |
| B. | LIST  | E. | RUN  |
| C. | LLIST | F. | SAVE |

- |       |    |                                                                                              |
|-------|----|----------------------------------------------------------------------------------------------|
| _____ | 1. | Causes the program currently in memory to be saved on secondary storage, probably a disk.    |
| _____ | 2. | Causes the printer to print all or part of the program currently in memory.                  |
| _____ | 3. | Causes the program identified to be retrieved from secondary storage and loaded into memory. |
| _____ | 4. | Clears primary storage.                                                                      |
| _____ | 5. | Causes the program in memory to be executed.                                                 |
| _____ | 6. | Causes all or part of the program currently in memory to be displayed on the CRT screen.     |

## PART I (Continued)

### Project 2

**INSTRUCTIONS:** Answer the following questions.

1. Define hierarchical grouping.
  
  
  
  
  
  
  
  
  
  
2. What are the duties of the chief programmer teams.
  
  
  
  
  
  
  
  
  
  
3. Define "egoless" programming teams.

## **PARTS II-IV: PROGRAMMING TECHNIQUES, EDITING, AND WRITING PROGRAMS**

### **Project 1**

**INSTRUCTIONS:** Answer the following questions.

1. Write a REM statement, giving the programmer's name.
  
  
  
  
  
  
  
  
  
  
2. What will the following PRINT statements print?
  - A. PRINT "HELLO","AND WELCOME"
  
  
  
  - B. PRINT "HELLO";" AND WELCOME"
  
  
  
  - C. PRINT "5 + 10 = 15"
  
  
  
  - D. PRINT 5 + 10

## PARTS II-IV (Continued)

### Project 2

**INSTRUCTIONS:** Load a high-level language. Key in the following program. Save the program under *Test1urs* (urs is your initials--example jcp).

```
30 PRINT " MY NAME IS CRT"  
40 PRINT " I AM A PROGRAMMER"  
50 PRINT " I MUST REMEMBER TO PRESS TAB AT THE END OF EACH  
   LINE I TYPE"
```

### Project 3

**INSTRUCTIONS:** Load the program you keyed in Project 2. Use your name in line 30, and correct the error in the last Print statement. Save as *Test2urs*.

### Project 4

**INSTRUCTIONS:** Key in the following program. Save it as *Test3urs*.

```
40 PRINT "THESE ARE MY GRADES"  
50 PRINT "80 90 77 80"  
60 PRINT "MY AVERAGE IS "
```

### Project 5

**INSTRUCTIONS:** Reload file *Test3urs*. Modify the program to show your last four grades. Modify line 60 to calculate and print the correct average. Save it as *Test4urs*.

## COMPETENCY TEST

### MODULE C: DOCUMENTATION, DEBUGGING AND TESTING

#### PART I: DOCUMENTATION

##### Project 1

**INSTRUCTIONS:** What is the difference between internal and external documentation?  
List four items often found in external documentation.

##### Project 2

**INSTRUCTIONS:** If you owned a business and had a programmer working for you, why would you insist on desk-checking?



## **PART II: ERROR TYPES**

**INSTRUCTIONS:** List the three types of errors and define each.

1.

2.

3.

## **PART III: TESTING**

**INSTRUCTIONS:** Develop a test for a program to keep student grades. When testing the program, what data will be examined as:

1. Typical data:

2. Unusual but valid data:

3. Incorrect or inappropriate data:

## COMPETENCY TEST

### MODULE D: USING ARITHMETIC

#### PART I: VARIABLES

##### Project 1

**INSTRUCTIONS:** List the rules for variable names.

##### Project 2

**INSTRUCTIONS:** List and define the two types of variables.

1)

2)

## PART II: ARITHMETIC OPERATORS

### Project 1

**INSTRUCTIONS:** List the hierarchy of operations.

1)

2)

3)

4)

5)

### Project 2

**INSTRUCTIONS:** Match the term to the proper symbol.

\_\_\_\_\_ 1. /

\_\_\_\_\_ 2. \*

\_\_\_\_\_ 3. +

\_\_\_\_\_ 4. \*\*

\_\_\_\_\_ 5. -

A. Exponentiation

B. Subtraction

C. Division

D. Addition

E. Multiplication

### PART III: EQUATIONS

**INSTRUCTIONS:** Fill in the blanks with the proper number or symbol for the following equations.

1.  $50 * 4 - 17 = \underline{\hspace{2cm}}$
2.  $\underline{\hspace{2cm}} + 20 / 5 = 34$
3.  $(20 * 2) + (20 / 4) - \underline{\hspace{2cm}} = 20$
4.  $14 \underline{\hspace{2cm}} 16 * 30 = 494$
5.  $5 ** 3 / 2 = \underline{\hspace{2cm}}$
6.  $10 \underline{\hspace{2cm}} 6 / 3 = 20$
7.  $\underline{\hspace{2cm}} / 5 + 20 * 4 = 120$
8.  $500 + 50 - 200 * 2 = \underline{\hspace{2cm}}$

### PART IV: PROGRAMMING WITH THE KEYWORD LET

**INSTRUCTIONS:** Write a short paragraph explaining how the keyword LET is used in a program and what its purpose is.

## COMPETENCY TEST

### MODULE E: INPUT CONCEPTS

#### PARTS I & II: READ/DATA, IF...THEN, GOTO, AND INPUT

##### Project 1

INSTRUCTIONS: What would be printed if you had the following data and format?

| <u>DATA</u>    | <u>FORMAT</u> | <u>RESULTS</u> |
|----------------|---------------|----------------|
| 1. 345.64      | ###.#         | _____          |
| 2. 2545.758    | #,###.##      | _____          |
| 3. 6540.90     | \$#,###.##    | _____          |
| 4. -243        | ###-          | _____          |
| 5. PROGRAMMING | \ \           | _____          |

##### Project 2

INSTRUCTIONS: What is the value for each of the variables on the specific attempt?

1. 10 READ A, R

.

80 DATA 65, 5.50, 45, 8.75

1ST TIME A = \_\_\_\_\_ R = \_\_\_\_\_

2ND TIME A = \_\_\_\_\_ R = \_\_\_\_\_

## PARTS I & II (Continued)

### Project 2 (Continued)

2. 10 READ C\$, S\$

80 DATA AUSIN, TX, NOEL, MO  
90 DATA TUSLA, OK

1ST TIME C\$ = \_\_\_\_\_ S\$ = \_\_\_\_\_

2ND TIME C\$ = \_\_\_\_\_ S\$ = \_\_\_\_\_

3RD TIME C\$ = \_\_\_\_\_ S\$ = \_\_\_\_\_

### Project 3

INSTRUCTIONS: Correct the errors in the following programs.

1. 100 READ N\$,AMT  
120 IF N\$="EOD" THEN 200  
130 PRINT N\$,AMT  
150 GOTO 100  
200 END  
1000 DATA 90,GEORGE,130,MARY,30.33,ALICE,EOD,0
2. 100 READ N\$,AMT  
120 IF N\$=EOD THEN 200  
130 PRINT AMT,N\$  
140 GOTO 100  
200 END  
1000 DATA SAM,55,RICHARD,77,TAYLOR,6,EOD
3. 100 READ N\$,AMT,TME  
120 IF N\$=999 THEN 200  
130 PRINT N\$,AMT,TME  
140 GOTO 100  
200 END  
1000 DATA JOE,55,2,MARY,4,3,SAM,123,2,5,BILL,332,5,999

### PART III: OUTPUT

#### Project 1

INSTRUCTIONS: Flowchart and code the following program: Input part number, quantity desired, cost each, and the tax rate. The output should appear as:

| Part<br>Number | Quantity<br>Required | Cost<br>Each | Tax<br>Amt. | Extension |
|----------------|----------------------|--------------|-------------|-----------|
| XXXX           | XX                   | XX.XX        | X.XX        | XX.XX     |

#### Project 2

INSTRUCTIONS: Flowchart and code a program using READ and DATA that will produce the following output.

| NAME          | PAY<br>RATE | HOURS<br>WORKED | WEEKLY<br>PAY |
|---------------|-------------|-----------------|---------------|
| =====         | =====       | =====           | =====         |
| TAMARA COX    | 7.75        | 40              | XXX.XX        |
| FRED JOHNSON  | 6.55        | 40              | XXX.XX        |
| DAVID BOOTH   | 12.00       | 38              | XXX.XX        |
| WILLIAM JONES | 15.00       | 22              | XXX.XX        |

## COMPETENCY TEST

### MODULE F: BRANCHING

#### PART I: RELATIONAL OPERATORS

**INSTRUCTIONS:** List the six relational operators.

- 1)
- 2)
- 3)
- 4)
- 5)

#### PART II: LOGICAL OPERATORS

**INSTRUCTIONS:** List the logical operators.

- 1)
- 2)
- 3)

#### PART III: CONDITIONAL AND UNCONDITIONAL BRANCHING

**INSTRUCTIONS:** Explain the difference between a conditional branch and an unconditional branch.



## PART IV: PROGRAMMING WITH CONDITIONAL BRANCHING

**INSTRUCTIONS:** Design, code, key, and execute a program to list the salesperson, number of miles traveled, and reimbursement.

If the miles are less than 1000, the fee is 10 cents a mile.

If the miles are equal to 1000 and less than 2000, the fee is 12 cents a mile.

If the miles are equal to 2000 and less than 3000, the fee is 14 cents a mile.

If the miles are over 3000, the fee is 16 cents a mile.

**Input:**

| Salesperson  | Miles |
|--------------|-------|
| Jerry Hill   | 2494  |
| Lousie Dawes | 3786  |
| Wendy James  | 695   |
| Sharon Aiken | 1768  |

Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## COMPETENCY TEST

### MODULE G: TOTALS

#### PART I: TOTALING AND COUNTING

**INSTRUCTIONS:** In your own words, define totaling and counting. Explain their differences.

#### PARTS II & III: SUBTOTALS, TOTALS, AND COUNTING

**INSTRUCTIONS:** Write a program to determine tuition for college students. For the data, have 5 students, indicate in- or out-of-state residence, and number of hours enrolled. Out-of-state students pay tuition of \$100 per hour, and in-state students pay tuition of \$165 per hour. The output should appear as:

| NAME                                  | RESIDENCE | HOURS | TUITION  |
|---------------------------------------|-----------|-------|----------|
| JOHNSON                               | I         | 12    | \$XXX.XX |
| TOTALS                                |           | XXX   | \$XXX.XX |
| TOTAL NUMBER OF IN-STATE STUDENTS     |           |       | XX       |
| TOTAL NUMBER OF OUT-OF-STATE STUDENTS |           |       | XX       |

Flowchart and code the program with the following data:

JOHNSON,I,12,BANKS,I,16,HAGGERTY,O,12,POWELL,I,16,MCDANIEL,O,10.

## COMPETENCY TEST

### MODULE H: MODULE PROGRAMMING

#### PART I: MODULAR PROGRAMMING

**INSTRUCTIONS:** In a short paragraph, define modular programming.

#### PART II: MAIN MODULES AND SUBMODULES

##### Project 1

**INSTRUCTIONS:** The following program contains a syntax error. Find the error, and rewrite the line of code to eliminate the error.

```
1. 10 REM THIS PROGRAM USES GOSUB
    .
    .
    .
    120 IF HRS > 40 THEN GOSUB 2000
    .
    .
    .
    2000 REM CALCULATE FOR OVERTIME
    2010 OTHR = HRS = 40
    2020 OTRATE = RATE * 1.5
    2030 OTPAY = OTHR * OTRATE
    3000 REM CALCULATE FOR DEDUCTIONS
    .
    .
```

## PART II (Continued)

### Project 2

**INSTRUCTION:** The following program contains a logic error. Find the error, and rewrite the line of code to eliminate the error.

1. 10 REM THIS IS A PROGRAM TO ILLUSTRATE GOSUB  
.  
.  
.  
100 CLS  
110 GOSUB 1000: REM GETS THE DATA  
120 GOSUB 2000: REM PRINT THE OUTPUT REPORT  
130 GOSUB 3000: REM DOES CALCULATIONS FOR THE REPORT  
140 GOTO 70  
150 END

## COMPETENCY TEST

### MODULE I: CONTROLLED LOOPS

#### Project 1

**INSTRUCTIONS:** Write a program that will create a table converting Fahrenheit to Celsius. Have the table show Fahrenheit degrees from 0 to 220. The formula for conversion is  $C = 5/9 * (F-32)$ . The output should appear as:

#### TABLE OF CORRESPONDING TEMPERATURES

FAHRENHEIT

CELSIUS

#### Project 2

**INSTRUCTIONS:** Flowchart and code the following program: Input the number of salespeople you are going to enter. Input the salesperson's name, rate of commission in decimal form, and amount of sales. Print the salesperson's name, rate of commission, amount of sales, and gross pay. Gross pay is determined by multiplying the amount of sales by the rate of commission. Total the gross pay and the sales for all of the salespeople entered. The output should appear as:

| NAME   | RATE OF<br>COMMISSION | AMOUNT OF<br>SALES | GROSS<br>PAY |
|--------|-----------------------|--------------------|--------------|
| TOTALS |                       | \$X,XXX.XX         | \$XXX.XX     |

What tests would be appropriate?

## COMPETENCY TEST

### MODULE J: ADVANCED CONCEPTS

#### PARTS I-IV: WRITING PROGRAMS

**INSTRUCTIONS:** Pseudocode and code the following inventory program. The program should first list all the parts in inventory. The program should then prompt to enter a part number. When the part number is entered, that part number and the price should be displayed. Use the following data.

| Part No. | Price  |
|----------|--------|
| 192888   | 8.90   |
| 320122   | 9.30   |
| 431289   | 2.20   |
| 392871   | .40    |
| 578654   | 101.09 |

## ANSWERS TO ACTIVITIES

### MODULE A: SETUP PROCEDURES

#### Activity 1

**Projects 1-2** Evaluated to the satisfaction of the instructor.

#### Activity 2

1. **Problem definition**--the first and most important step in problem solving is understanding what the problem is.
2. **Information desired**--decide what information is needed and the desired form of output.
3. **Data required**--determine what data are needed to provide the desired information, how they will be accumulated, and what form and by what method the data should be entered.
4. **Processing needed**--resolve how to process the data to produce the desired results; know what calculations will be required and whether the programmer is familiar with the algorithm necessary to do the processing or will need additional help or research.

### MODULE B: INTRODUCTION TO PROGRAMMING

#### Activity 1

- |    |   |     |   |
|----|---|-----|---|
| 1. | A | 6.  | B |
| 2. | A | 7.  | B |
| 3. | A | 8.  | B |
| 4. | B | 9.  | A |
| 5. | B | 10. | B |

#### Activity 2

**Program 1** MY NAME IS JERRY  
THE PERSON NEXT TO ME IS NANCY

**Program 2** 5 + 5 + 10  
IS 20

**Activity 2 (Continued)**

**Program 3**    THE GRADES FOR ALICE ARE  
                 90 80 70 86  
                 THE AVERAGE FOR ALICE IS 109

**Activity 3**    Students should be able to independently load the high-level language being used in the class. The program should be coded so that there is no error message and the correct output appears on the screen. The students should be able to save the program with a legitimate file name.

**Projects 1-3**   Evaluated to the satisfaction of the instructor.

**Activity 4**

**Project 1**

1.    HI!  
      WELCOME TO THE WORLD OF PROGRAMMING
2.    HI! WELCOME TO THE WORLD OF PROGRAMMING
3.    THE                    END
4.    THE END
5.    THE END

**Project 2**

```
10 REM **** PROGRAMMER'S NAME ****
20 REM **** DATE ****
30 PRINT " THE NAME OF MY COMPUTER TEACHER"
40 PRINT "                    IS"
50 PRINT "                    MR. JEFFERSON"
```

**Project 3**

Line 50 on Project 2 will change to instructor's name.

**Project 4**

```
10 PRINT "THIS WILL APPEAR";
20 PRINT "ON ONE LINE"
```



**Project 5**

- a. 50 PRINT "NOTICE", "THE", "EFFECTS", "OF", "COMMAS"
- b. 50 PRINT "YOU CAN ALSO"  
60 PRINT  
70 PRINT "SKIP LINES"

**MODULE C: TESTING, DEBUGGING AND DOCUMENTATION**

**Activity 1**

**Project 1-2** Evaluated to the satisfaction of the instructor.

**Activity 2** Evaluated to the satisfaction of the instructor.

**Activity 3**

**Projects 1-2** Evaluated to the satisfaction of the instructor.

**MODULE D: USING ARITHMETIC**

**NOTE:** The answers given are one possible way of coding the program. The student should include REMS which the teacher deems appropriate.

**Activity 1**

- |    |   |     |   |
|----|---|-----|---|
| 1. | A | 6.  | C |
| 2. | C | 7.  | A |
| 3. | D | 8.  | B |
| 4. | B | 9.  | D |
| 5. | B | 10. | D |

**Activity 2**

**Project 1**

- 1. PRINT 10.50 + 27.26 + 36.10
  - 2. PRINT (90 + 80 + 70 + 95)/4
  - 3. PRINT 260 \* 10
  - 4. PRINT (9.10 + 12.24 + 5) \* .055
- 467

**Activity 2 (Continued)**

**Project 2**

1. 50 PRINT 100 + 200 + 5000 + 250
2. 60 PRINT  $9^2$ ,  $123^2$
3. 70 PRINT  $(5 + 6 + 9)/4$
4. 80 PRINT  $(5.80 + 40.90 + 3.27) * .0525$

**Project 3**

50 PRINT  $(8+7+6+4)*5.20$

**Activity 3**

1. 500
2. 158.00
3. 207
4. 8975
5. 990

**Activity 4**

**Project 1**

1. LET A = 0
2. LET A = X/Y

**Project 2**

```
50 REM THIS PROGRAM WILL AVERAGE THREE NUMBERS
60 REM A = NUMBER 1
70 REM B = NUMBER 2
80 REM C = NUMBER 3
90 REM G = AVERAGE
95 REM TO STOP PROGRAM ENTER 0 FOR INPUT A
100 INPUT A, B, C
110 IF A = 0 THEN 200
120 LET G = (A+B+C)/3
170 PRINT "AVERAGE IS: "; G
180 GOTO 100
200 END
```

## MODULE E: INPUT STATEMENTS

### Activities 1-2

#### Project 1

- |    |          |     |           |
|----|----------|-----|-----------|
| 1. | 423.7    | 6.  | \$78.78   |
| 2. | .75      | 7.  | \$247.95  |
| 3. | 9.68     | 8.  | 518       |
| 4. | 45.75    | 9.  | -734      |
| 5. | \$120.90 | 10. | THIS IS T |

#### Project 2

- |    |         |          |                        |
|----|---------|----------|------------------------|
| 1. | 35      | 4.50     |                        |
|    | 40      | 4.75     |                        |
| 2. | 5       | 9        | 4                      |
|    | 7       | 8        | 1                      |
|    | 3       | 6        | WILL GET ERROR MESSAGE |
| 3. | JESSICA | 35.62    |                        |
|    | NATHAN  | 32.45    |                        |
|    | SHAWN   | 15.62    |                        |
| 4. | 459     | NOTEBOOK |                        |
|    | 123     | PENCIL   |                        |

#### Project 3

1. 10 READ A, N\$
2. 10 READ N, C\$  
50 DATA 1000, JONES  
60 DATA 2000, SMITH

#### Project 4

1. 20 IF D\$ = "EOD" THEN 70
2. 30 PRINT I\$, P  
40 GOTO 10

**BCPI/Program Development**  
**Unit 2: Programming Concepts and Applications**  
**Teacher's Guide/ANSWERS**

**Project 5**

1. 90 DATA EOD, 0, 0
2. 60 DATA DESK, 450.00, 345.00  
70 DATA CHAIR, 125.00, 75.00  
80 DATA LAMP, 75.00, 58.00  
90 DATA EOD, 0, 0

**Project 6**

1. 30 A = L\*W  
40 PRINT L,W,A
2. 50 GOTO 10

**Activity 3**

**Project 1** Evaluated to the satisfaction of the instructor.

**Project 2**

1. 50 INPUT "PERSON YOU ARE WRITING ";N\$  
60 INPUT "THANK YOU SO MUCH FOR THE "; I\$  
70 INPUT "IT WAS VERY NICE OF YOU. I RECEIVED IT "; R\$  
80 INPUT "LOOKING FORWARD TO SEEING YOU ";S\$  
90 CLS  
100 PRINT "Dear ";N\$;","  
110 PRINT  
120 PRINT "Thank you so much for the ";I\$;". ";  
130 PRINT " It was very nice of you. I received it ";R\$;". ";  
140 PRINT " I am looking forward to seeing you ";S\$;". "  
150 PRINT:PRINT  
160 PRINT " Sincerely,"  
170 PRINT:PRINT  
180 PRINT " Your name"
2. 50 INPUT "ENTER A NUMBER ";NUM1  
60 INPUT "ENTER ANOTHER NUMBER ";NUM2  
70 PRINT  
80 PRINT "IF THE TWO NUMBERS YOU ENTER ARE  
MULTIPLIED TOGETHER, WHAT WOULD THE ANSWER BE?"  
90 PRINT  
100 INPUT "ENTER YOUR ANSWER ";A  
110 PRINT  
120 PRINT "IF YOU GUESSED ";NUM1\*NUM2;"," " YOU ARE RIGHT."

### Project 3

1. 10 INPUT "ENTER STUDENT NAME: "; S\$  
20 INPUT "ENTER TEST SCORE 1: "; TEST1  
30 INPUT "ENTER TEST SCORE 2: "; TEST2  
70 INPUT "DO YOU HAVE MORE STUDENTS (Y/N) "; R\$

Students could use a combination of PRINT and INPUT statements.

2. 60 DATA 25.00, .25, 49.00, .18, 0, 0

### Project 4

1. 30 PAY = H \* R
2. 50 GOTO 10

#### PROGRAM 1

```
5 CLS
10 PRINT "SALESPERSON LIST"
20 PRINT:PRINT
30 PRINT "SALESPERSON
  SALESPERSON"
40 PRINT " NUMBER NAME
  DEPARTMENT"
50 PRINT
60 IS = " ### \ \ \ \"
70 READ D$, N, N$
80 IF D$ = "EOD" THEN 170
90 PRINT USING IS; N, N$, D$
100 GOTO 70
110 DATA HARDWARE, 349, STEVE
  WILLS
120 DATA SPORTING GOODS, 153,
  JERRY HILL
130 DATA SHOES, 279, TAMMY
  MOORE
140 DATA TOYS, 480, WENDY DAVIS
150 DATA VIDEO, 784, PAM
  ANDERSON
160 DATA EOD, 999, EOD
170 END
```

#### PROGRAM 2

```
10 CLS
20 PRINT "SALES REPORT"
30 PRINT:PRINT
40 PRINT "SALESPERSON YTD
  CURRENT TOTAL"
50 PRINT " NAME SALES SALES
  SALES"
60 PRINT
70 IS = "\ \ #,###.## #,###.##
  ##,###.## "
80 READ N$, YS, CS
90 IF N$ = "EOD" THEN 190
100 TS = YS + CS
110 PRINT USING IS; N$, YS, CS, TS
120 GOTO 80
130 DATA EVEY HAYES, 4500.00,
  1000.00
140 DATA SUSAN MAPLE, 6700.00,
  2500.00
150 DATA ROBERT BATES, 5495.00,
  2750.00
160 DATA NANCY AUSTIN, 8945.00,
  1656.00
170 DATA DON WEST, 3456.00, 2340.00
180 DATA EOD, 0, 0
190 END
```

### PROGRAM 3

```
10 CLS
20 PRINT "SALES REPORT"
30 PRINT:PRINT
40 PRINT "SALESPERSON YTD
  CURRENT SALES TOTAL"
50 PRINT " NAME SALES SALES
  RETURNS SALES"
60 PRINT
70 IS = "\ \ ###.## ###.##
  ###.## ###.## "
80 READ N$, YS, CS, SR
90 IF N$ = "EOD" THEN 190
100 TS = YS + CS - SR
110 PRINT USING IS;N$, YS, CS, SR, TS
120 GOTO 80
130 DATA EVEY HAYES, 4500.00,
  1000.00, 575.00
140 DATA SUSAN MAPLE, 6700.00,
  2500.00, 1010.00
150 DATA ROBERT BATES, 5495.00,
  2750.00, 1750.00
160 DATA NANCY AUSTIN, 8945.00,
  1656.00, 56.00
170 DATA DON WEST, 3456.00,
  2340.00, 325.00
180 DATA EOD, 0, 0, 0
190 END
```

### PROGRAM 4

```
10 CLS
20 LPRINT "SALES COMMISSION
  REPORT"
30 LPRINT:LPRINT
40 LPRINT "SALESPERSON
  CURRENT COMMISSION
  COMMISSION"
50 LPRINT " NAME SALES RATE
  AMOUNT"
60 LPRINT
70 IS = "\ \ ###.## ##%
  ###.## "
80 INPUT "ENTER SALESPERSON
  NAME (EOD TO END) ";N$
90 IF N$ = "EOD" THEN 230
100 INPUT "ENTER CURRENT SALES
  ";SALES
110 INPUT "ENTER COMMISSION
  RATE (i.e., .06) ";RATE
120 COMAMT = SALES * RATE
130 RATE = RATE * 100
190 LPRINT USING IS;N$, SALES,
  RATE, COMAMT
220 GOTO 80
230 END
```

## MODULE F: BRANCHING

### Activity 1

#### Project 1

1. Equal to
2. Less than or equal to
3. Not equal to
4. Greater than
5. Less than
6. Greater than or equal to

#### Project 2

1. IF A\$ = "21" THEN COST = RATE + 250.00  
or  
IF A = 21 THEN COST = RATE + 250.00
2. IF T >= 50.00 THEN 90
3. IF RATE > .10 THEN 200
4. IF A > 250.00 OR A < 50.00 THEN 100
5. IF C IS GREATER THAN 99.99 PRINT "COST IS TOO HIGH-- ERROR"
6. IF CHOICE = 1 THEN 1000
7. IF P\$ < "A" THEN 250 or IF P < A THEN 250
8. IF N\$ = "EOD" THEN 400
9. IF R\$ = "YES" OR R\$ = "NO" THEN 460
10. IF AMT > 50.00 THEN 50

### Activity 2

1. AND, OR, NOT
2. Evaluated to the satisfaction of the instructor.

### Activity 3

- Project 1 Evaluated to the satisfaction of the instructor.

### Activity 3 (Continued)

#### Project 2

1. 50  
56
2. 3 NUMBERS OVER 100  
4 NUMBERS UNDER 100
3. 7 7 144

#### PROGRAM 1

```

10 CLS
20 PRINT " SALES QUOTA REPORT"
30 PRINT:PRINT
40 PRINT "SALESPERSON
CURRENT"
50 PRINT " NAME SALES"
60 PRINT:PRINT
70 IS = "\ \ #,###.##"
80 READ NAM$, SALES
90 IF NAM$ = "EOD" THEN 240
100 IF SALES <= 5000! THEN 80
110 PRINT USING IS; NAM$, SALES
120 GOTO 80
130 DATA LYNN HOLMAN, 5409.00
140 DATA SUSAN CUMMINGS,
4989.75
150 DATA BARBARA FARIN, 3458.34
160 DATA JEFF PAUL, 5000.00
170 DATA WALTER ANDERSON,
8794.98
180 DATA SONDR A HILL, 2050.99
190 DATA LORI ROSEN, 7945.24
200 DATA WES THOMAS, 6885.49
210 DATA JERRY STOOPS, 1495.95
220 DATA ALVIN BELL, 2567.89
230 DATA EOD, 0
240 END

```

#### PROGRAM 2

```

10 CLS
20 PRINT TAB(25); "SALES QUOTA
REPORT"
30 PRINT:PRINT
40 PRINT "SALESPERSON
CURRENT MONTHS AMOUNT"
50 PRINT " NAME SALES
EMPLOYED BELOW QUOTA"
60 PRINT:PRINT
70 IS = "\ \ #,###.## ##
#,###.##"
80 READ NAM$, SALES, MONTHS
90 IF NAM$ = "EOD" THEN 250
100 IF MONTHS <= 8 THEN 80
110 IF SALES > 5000! THEN 80 ELSE
BELOW = 5000! - SALES
120 PRINT USING IS; NAM$, SALES,
MONTHS, BELOW
130 GOTO 80
140 DATA LYNN HOLMAN, 5409.00,19
150 DATA SUSAN CUMMINGS,
4989.75,15
160 DATA BARBARA FARIN,
3458.34,9
170 DATA JEFF PAUL, 5000.00,5
180 DATA WALTER ANDERSON,
8794.98,24
190 DATA SONDR A HILL, 2050.99,13
200 DATA LORI ROSEN, 7945.24,20
210 DATA WES THOMAS, 6885.49,18
220 DATA JERRY STOOPS, 1495.95,10
230 DATA ALVIN BELL, 2567.89,3
240 DATA EOD, 0,0
250 END

```



1000

440

## MODULE G: TOTALS

### Activity 1

Totaling--adding up a series of numbers and producing a total or sum. A variable is assigned as an accumulator; then each new amount is added to the number already in the accumulator.

Counting--variables are incremented by one(or by whatever number you choose to increment the counter). A count will show how many items have been processed.

### PROGRAM 1

```
10 CLS
20 PRINT "      SALES QUOTA REPORT"
30 PRINT:PRINT
40 PRINT "SALESPERSON      CURRENT"
50 PRINT " NAME          SALES"
60 PRINT:PRINT
70 IS = "\      \      #,###.##"
80 READ NAM$, SALES
90 IF NAM$ = "EOD" THEN 260
100 IF SALES <= 5000! THEN 80
110 TSALES = TSALES + SALES
120 TNAM = TNAM + 1
130 PRINT USING IS; NAM$, SALES
140 GOTO 80
150 DATA LYNN HOLMAN, 5409.00
160 DATA SUSAN CUMMINGS, 4989.75
170 DATA BARBARA FARIN, 3458.34
180 DATA JEFF PAUL, 5000.00
190 DATA WALTER ANDERSON, 8794.98
200 DATA SONDR A HILL, 2050.99
210 DATA LORI ROSEN, 7945.24
220 DATA WES THOMAS, 6885.49
230 DATA JERRY STOOPS, 1495.95
240 DATA ALVIN BELL, 2567.89
250 DATA EOD, 0
260 PRINT:PRINT
270 PRINT "TOTAL SALESPERSON "; TNAM
280 PRINT USING "TOTAL CURRENT SALES ABOVE QUOTA
    $###,###.##"; TSALES
290 END
```

## PROGRAM 2

```

10 CLS
20 PRINT TAB(25);"SALES QUOTA "
   REPORT"
30 PRINT:PRINT
40 PRINT "SALESPERSON
   CURRENT MONTHS AMOUNT"
50 PRINT " NAME SALES
   EMPLOYED BELOW QUOTA"
60 PRINT:PRINT
70 IS = "\ \ ###.## ##
   ###.##"
80 READ NAM$, SALES, MONTHS
90 IF NAM$ = "EOD" THEN 270
100 TOTALS = TOTALS + 1
110 IF MONTHS <= 8 THEN 80
120 IF SALES > 5000! THEN 80 ELSE
   BELOW = 5000! - SALES
130 TOTALBQ = TOTALBQ + 1
140 PRINT USING IS; NAM$, SALES,
   MONTHS, BELOW
150 GOTO 80
160 DATA LYNN HOLMAN, 5409.00,19
170 DATA SUSAN CUMMINGS,
   4989.75,15
180 DATA BARBARA FARIN,
   3458.34,9
190 DATA JEFF PAUL, 5000.00,5
200 DATA WALTER ANDERSON,
   8794.98,24
210 DATA SONDR A HILL, 2050.99,13
220 DATA LORI ROSEN, 7945.24,20
230 DATA WES THOMAS, 6885.49,18
240 DATA JERRY STOOPS, 1495.95,10
250 DATA ALVIN BELL, 2567.89,3
260 DATA EOD, 0,0
270 PRINT:PRINT
280 PRINT "TOTAL NUMBER OF
   SALESPERSONS "; TOTALS
290 PRINT "TOTAL NUMBER OF
   SALESPERSONS BELOW QUOTA ";
   TOTALBQ
300 END

```

## PROGRAM 3

```

10 CLS
20 PRINT TAB(15);"ACCOUNTS
   RECEIVABLE REGISTER"
30 PRINT
40 PRINT "CUSTOMER CUSTOMER
   INVOICE"
50 PRINT " NUMBER NAME
   AMOUNT"
60 PRINT:PRINT
70 DET$ = " #### \ \
   ###.##"
80 TOTSUB$ = " CUSTOMER TOTAL
   ###.##*"
86 TOTFIN$ = " GRAND TOTAL
   ###.##*"
90 READ NUMB, NAM$, AMT
100 IF NUMB = 0 THEN 280
110 IF NUMB <> COMP AND COMP
   > 0 THEN PRINT USING
   TOTSUB$;SUBTOT: SUBTOT =
   0:PRINT
140 SUBTOT = SUBTOT + AMT
150 FINTOT = FINTOT + AMT
160 COMP = NUMB
165 PRINT USING DET$; NUMB,
   NAM$, AMT
170 GOTO 90
180 DATA 4326, CURRIE
   COMPUTERS, 450.87
190 DATA 4326, CURRIE
   COMPUTERS, 54.00
200 DATA 4326, CURRIE
   COMPUTERS, 35.00
210 DATA 7689, JARRETT
   SOFTWARE, 234.65
220 DATA 7689, JARRETT
   SOFTWARE, 87.00
230 DATA 7689, JARRETT
   SOFTWARE, 123.90
240 DATA 7689, JARRETT
   SOFTWARE, 67.83
250 DATA 1539, WJ WAREHOUSE,
   342.00
260 DATA 1539, WJ WAREHOUSE,
   653.00
270 DATA 0, EOD, 0
280 PRINT USING TOTSUB$; SUBTOT
285 PRINT
290 PRINT USING TOTFIN$; FINTOT
300 END

```

## MODULE H: MODULE PROGRAMMING

### PROGRAM 1

```
10 CLS
20 PRINT TAB(18);"HILL PRINTING COMPANY PAYROLL REPORT"
30 PRINT:PRINT
40 PRINT "  EMPLOYEE    HOURS  RATE OF  GROSS"
50 PRINT "    NAME      WORKED   PAY     PAY"
60 PRINT:PRINT
70 I$ = "\      \    ##    ##.##  #,###.##"
80 E$ = "\      \    ##  ERROR -- TOO HOURS REPORTED"
100 REM ***** MAIN MODULE
110 GOSUB 200
120 IF NAM$ = "EOD" THEN 700
130 GOSUB 300
140 GOSUB 400
150 GOSUB 500
160 GOTO 100
200 REM ***** GET DATA
210 READ NAM$, HOURS, RATE
299 RETURN
300 REM ***** CHECK DATA
310 IF HOURS > 40 THEN FLAG$ = "ERROR"
399 RETURN
400 REM ***** CALCULATE
410 GROSS = HOURS * RATE
499 RETURN
500 REM ***** PRINT OUTPUT
510 IF FLAG$ = "ERROR" THEN PRINT USING E$; NAM$, HOURS ELSE PRINT
    USING I$; NAM$, HOURS, RATE, GROSS
520 FLAG$ = ""
599 RETURN
600 DATA CODY MACKIE, 39, 4.90, ZACK CRAWFORD, 40, 6.95
610 DATA CHUCK DUKE, 42, 4.50, TRAVIS EVERS, 40, 5.90
620 DATA TRACY STEELMAN, 45, 4.00, BETH FORD, 40, 4.00
630 DATA EOD, 0, 0
700 END
```

## PROGRAM 2

```
10 CLS
20 PRINT TAB(18);"HILL PRINTING COMPANY PAYROLL REPORT"
30 PRINT:PRINT
40 PRINT "  EMPLOYEE      REGULAR  OVERTIME  INCOME  NET"
50 PRINT "    NAME          PAY      PAY      TAXES   PAY"
60 PRINT:PRINT
70 IS = "\      \  ###.##  ###.##  ###.##  ###.##"
100 REM ***** MAIN MODULE
110 GOSUB 200
120 IF NAM$ = "EOD" THEN 700
140 GOSUB 400
150 GOSUB 500
160 GOTO 100
200 REM ***** GET DATA
210 READ NAM$,HOURS, RATE
299 RETURN
400 REM ***** CALCULATE
410 IF HOURS > 40 THEN REGPAY = 40 * RATE ELSE REGPAY = HOURS
    *RATE
420 IF HOURS > 40 THEN OVERTIME = (HOURS-40) * RATE * 1.5
    ELSE OVERTIME = 0
430 TAXES = (REGPAY + OVERTIME) * .05
440 NETPAY = REGPAY + OVERTIME - TAXES
450 TEMP = TEMP + 1
460 TREGPAY = TREGPAY + REGPAY
470 TOVERTIME = TOVERTIME + OVERTIME
480 TTAXES = TTAXES + TAXES
490 TNETPAY = TNETPAY + NETPAY
499 RETURN
500 REM ***** PRINT OUTPUT
510 PRINT USING IS; NAM$, REGPAY, OVERTIME, TAXES, NETPAY
599 RETURN
600 DATA CODY MACKIE, 39, 4.85, ZACK CRAWFORD, 40, 6.95
610 DATA CHUCK DUKE, 42, 4.50, TRAVIS EVERS, 40, 5.90
620 DATA TRACY STEELMAN, 45, 4.00, BETH FORD, 40, 4.00
630 DATA EOD, 0, 0
700 REM ***** TOTAL LINE
710 PRINT
720 T$ = "TOTALS"
730 PRINT USING IS;T$,TREGPAY, TOVERTIME, TTAXES, TNETPAY
740 PRINT "TOTAL EMPLOYEES ";TEMP
750 END
```

## MODULE I: CONTROLLED LOOPS

### Activity 1

#### Project 1

1.     1     4     7     10     13     16     19
2.     30     27     24     21     18     15     12     9     6     3
3.     3     1  
        3     2  
        3     3  
        3     4  
        2     1  
        2     2  
        2     3  
        2     4  
        1     1  
        1     2  
        1     3  
        1     4

#### Project 2

1.     10 FOR T = 40 TO 20 STEP -1  
        Step could be any negative number
2.     70 NEXT C  
        80 NEXT B
3.     40 IF K > 9 THEN 80  
        50     FOR G = 1 TO 1000:NEXT G  
        80 PRINT "K = ";K

### PROGRAM

```
10 CLS
20 INPUT "WHAT IS THE COST OF ONE ITEM? ";COST
25 PRINT:PRINT
30 PRINT "  FUND-RAISING PRICE LIST"
40 PRINT USING "  FOR AN ITEM COSTING ###.##";COST
50 PRINT:PRINT
60 PRINT "  NUMBER OF      TOTAL"
70 PRINT "      ITEMS      PRICE"
80 PRINT
90 I$ = "  ##      ###.##"
120 FOR J = 1 TO 10
130 PRICE = J * COST
140 PRINT USING I$; J, PRICE
150 NEXT J
155 PRINT
160 INPUT "DO YOU HAVE ANOTHER ITEM WHICH NEEDS A TABLE RUN
(Y/N)? ";R$
170 IF R$ = "Y" THEN 10
180 IF R$ = "N" THEN END
190 PRINT "INVALID RESPONSE"
200 GOTO 160
```

### MODULE J: ADVANCED CONCEPTS

#### Activity 1

1.     40 READ N\$(X)  
       60 FOR A = 1 TO N  
       70 PRINT N\$(A)
  
2.     40 J = 5  
       50 DIM R\$(J)  
       55 FOR B = 1 TO J  
       60 READ N\$(B)  
       65 NEXT B  
       70 FOR B = 1 TO J  
       80 PRINT N\$(B)  
       90 NEXT B  
       1000 DATA BUD, CATHY, JIM, MARGARET, SISSY



PROGRAM

```
10 CLS
20 DIM N$(10),C$(10)
30 FOR N = 1 TO 10
40 READ N$(N),C$(N)
50 NEXT N
60 N = N-1
90 CLS
100 PRINT "WHAT TYPE OF REPORT DO YOU WANT?"
105 PRINT:PRINT
110 PRINT "1 ... ALPHABETIZED LISTING OF CONTESTANTS"
120 PRINT "2 ... ALPHABETIZED LISTING BY EVENT"
130 PRINT "3 ... TERMINATE PROGRAM"
140 PRINT:PRINT
150 INPUT "PLEASE ENTER CHOICE ";C
160 IF C = 1 THEN GOSUB 300
170 IF C = 2 THEN GOSUB 500
180 IF C = 3 THEN END
190 IF C = 1 THEN GOSUB 700 ELSE GOSUB 800
200 GOTO 90
300 REM ***** SORT BY NAME
310 FOR X = 1 TO N-1
320 FOR Y = X+1 TO N
330 IF N$(X) < N$(Y) THEN 360
340 T$ = N$(X): N$(X) = N$(Y): N$(Y) = T$
350 T$ = C$(X): C$(X) = C$(Y): C$(Y) = T$
360 NEXT Y
370 NEXT X
380 RETURN
500 REM ***** SORT BY CONTEST
510 FOR X = 1 TO N-1
520 FOR Y = X+1 TO N
530 IF C$(X) < C$(Y) THEN 560
540 SWAP N$(X), N$(Y)
550 SWAP C$(X), C$(Y)
560 NEXT Y
570 NEXT X
580 RETURN
700 REM ***** PRINT REPORT BY CONTESTANT NAME
710 CLS
720 PRINT TAB(15);"CONTESTANT LISTING"
730 PRINT:PRINT
740 PRINT "CONTESTANT          CONTEST"
750 PRINT
760 IS = "\          \          \"
770 FOR Z = 1 TO 10
780 PRINT USING IS; N$(Z), C$(Z)
790 NEXT Z
798 W$ = INKEY$: IF W$ = "" THEN 898
799 RETURN
```

417



### PROGRAM (Continued)

```
800 REM ***** PRINT REPORT BY CONTESTANT NAME
810 CLS
820 PRINT TAB(15);"CONTEST LISTING"
830 PRINT:PRINT
840 PRINT "CONTEST          CONTESTANT"
850 PRINT
860 IS = "\      \      \      \"
870 FOR X = 1 TO 10
880 PRINT USING IS; C$(X), N$(X)
890 NEXT X
898 W$ = INKEY$: IF W$ = "" THEN 798
899 RETURN
900 DATA "REA, SHELLY", "DATA ASSISTANT"
910 DATA "WILSON, TOMMY", "COMPUTER SPECIALIST"
920 DATA "MAHAN, TODD", "COMPUTER SPECIALIST"
930 DATA "KING, BRYAN", "DATA SPECIALIST"
940 DATA "ALEXANDER, ROBIN", "DATA ASSISTANT"
950 DATA "CASSELL, CINDY", "DATA SPECIALIST"
960 DATA "WESTER, TODD", "FINANCIAL SPECIALIST"
970 DATA "WAMPLER, BRENT", "COMPUTERIZED ACCOUNTING"
980 DATA "TOLETT, CHRIS", "PREPARED VERBAL"
990 DATA "KEENE, TAMMY", "FINANCIAL SPECIALIST"
```

### COMPETENCY TEST ANSWERS

#### MODULE A: SETUP PROCEDURES

Parts I-II      Evaluated to the satisfaction of the instructor.

#### MODULE B: INTRODUCTION TO PROGRAMMING

##### Part I

##### Project 1

- |    |    |    |   |
|----|----|----|---|
| 1. | F  | 4. | A |
| 2. | C  | 5. | E |
| 3. | D. | 6. | B |

**Part I (Continued)**

**Project 2**

1. Hierarchical grouping--a programming manager assigns the coding to another individual and does not actually participate in the coding task(s).
2. Chief programmer teams--a chief programmer is responsible for the project and the primary writer of code. Others on the team back up and support the chief programmer by writing code and locating needed documents.
3. "Egoless" programming teams--there is no designated chief, and assignments are determined in a democratic way with each member doing the part of the work for which he/she is best suited.

**Parts II-IV**

**Project 1**

1. Evaluated to the satisfaction of the instructor.
2. A. HELLO AND WELCOME  
B. HELLO AND WELCOME  
C.  $5 + 10 = 15$   
D. 15

**Project 2**

MY NAME IS CRT  
I AM A PROGRAMMER  
I MUST REMEMBER TO PRESS TAB AT THE END OF EACH LINE I TYPE

**Project 3**

MY NAME IS \_\_\_\_\_  
I AM A PROGRAMMER  
I MUST REMEMBER TO PRESS ENTER AT THE END OF EACH LINE I TYPE

- Project 4** Evaluated to the satisfaction of the instructor.

**Part II-IV (Continued)**

**Project 5**

|                             |                                |
|-----------------------------|--------------------------------|
| LINE 60--Will be changed to | 60 PRINT "MY AVERAGE IS ";     |
| LINE 70--Will be added      | 70 PRINT (80 + 90 + 77 + 80)/4 |

THESE ARE MY GRADES  
80 90 77 80  
MY AVERAGE IS 82

**MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING**

**Part I**

**Project 1**

Internal documentation is located within the program itself and can only be seen when the program is listed. External documentation is the written explanation that accompanies the program. Any four of the following would be correct.

1. Purpose of the software
2. Capabilities and limitations of the software
3. Operator instructions
4. Description of the program
5. Error messages

**Project 2**

Desk-checking is a careful examination and manual tracing of the program before it is coded. Desk-checking points out errors and possible problems. If you were paying an employee, you would want the overall time of coding a program to be as short as possible. Time spent in desk-checking more than compensates for time that would have to be spent correcting or possibly recording the program.

**Part II**

1. Run-time--an error that causes program execution to stop prematurely.
2. Syntax--mistakes in the programming language being used.
3. Logic--errors in the basic logic, or construction, of the program.

### Part III

1. Typical data--grades between 0 and 100
2. Unusual but valid data--grades between -.20 and 120 or decimal numbers.
3. Incorrect or inappropriate data: words instead of numbers, very high or low numbers (e.g., 1020 or -33)

## MODULE D: USING ARITHMETIC

### Part I

#### Project 1

1. May contain a number but cannot start with a number.
2. May not be the same as a keyword or a command.
3. May not contain a special symbol.
4. Should give an indication of what the variable represents.

#### Project 2

1. Numeric--used to represent a number.
2. Non-numeric or string--used to represent words, letters, or combinations of letters and numbers. String variables are written with a "\$" as the last character.

### Part II

#### Project 1

1. Operations inside parentheses
2. Exponentiation
3. Multiplication and division
4. Addition and subtraction
5. Operations performed from left to right

#### Project 2

1. C
2. E
3. D
4. A
5. B

451

BCPI/Program Development  
Unit 2: Programming Concepts and Applications  
Teacher's Guide/ANSWERS

Part III

- |    |     |    |      |
|----|-----|----|------|
| 1. | 183 | 5. | 62.5 |
| 2. | 30  | 6. | *    |
| 3. | 25  | 7. | 200  |
| 4. | +   | 8. | 150  |

Part IV      Evaluated to the satisfaction of the instructor.

MODULE E: INPUT CONCEPTS

Parts I-II

Project 1

1. 345.6
2. 2,545.76
3. \$6,540.90
4. 243-
5. PROGRAM

Project 2

1.    A = 65      R = 5.50  
      A = 45      R = 8.75
2.    C\$ = AUSTIN      S\$ = TX  
      C\$ = NOEL      S\$ = MO  
      C\$ = TULSA      S\$ = OK

Project 3

1.    1000 DATA GEORGE, 90, MARY, 130, ALICE, 30.33,EOD, 0
2.    120 IF N\$ = "EOD" THEN 200  
      1000 DATA SAM, 55, RICHARD, 77, TAYLOR, 6, EOD, 0
3.    120 IF N\$ = "999" THEN 200  
      1000 DATA JOE, 55, 2, MARY, 4, 3, SAM, 123, 2,BILL, 332, 5, 999, 0, 0

### Part III

#### Project 1

```

40 REM PN ..... PART NUMBER
50 REM QUAN ..... QUANTITY REQUIRED
60 REM COST ..... COST
70 REM TAX ..... TAX RATE
80 REM EXT ..... EXTENSION
90 REM TAMT ..... TAX IN DOLLARS
100 CLS
110 INPUT "ENTER PART NUMBER ";PN
120 INPUT "ENTER QUANTITY ";QUAN
130 INPUT "ENTER COST EACH ";COST
140 INPUT "ENTER TAX RATE ";TAX
150 TAMT = TAX*QUAN*COST
160 EXT = (QUAN*COST) + TAMT
200 PRINT "  PART      QUANTITY    COST      TAX  EXTENSION"
210 PRINT "NUMBER    REQUIRED    EACH      AMT"
220 PRINT "  PN"      "QUAN"      "COST"    "TAMT"  "EXT"

```

#### Project 2

```

40 REM N$ ..... NAME
50 REM PR..... PAY RATE
60 REM HW ..... HOURS WORKED
70 REM WP..... WEEKLY PAY
110 PRINT "      PAY      HOURS      WEEKLY"
120 PRINT "NAME    RATE    WORKED    PAY"
130 PRINT STRING$(70,"-")
200 READ N$, PR, HW
210 IF N$ = "EOD" THEN 300
220 WP = HW * PR
230 PRINT USING "\    \ ###.##  ##    #,###.##"
240 GOTO 200
300 END
1000 DATA TAMARA COX, 7.75, 40, FRED JOHNSON, 6.55, 40
1010 DATA DAVID BOOTH, 12.00, 38, WILLIAM JONES, 15.00, 22
1020 DATA EOD 0, 0

```

### MODULE F: BRANCHING

#### Part I

- |    |   |              |    |     |                        |
|----|---|--------------|----|-----|------------------------|
| 1. | = | Equal to     | 4. | < > | Not equal to           |
| 2. | < | Less than    | 5. | < = | Less than or equal to  |
| 3. | > | Greater than | 6. | > = | Greater to or equal to |

**Part II**

1. AND
2. OR
3. NOT

**Part III** Evaluated to the satisfaction of the instructor.

**Part IV**

```
10 CLS
20 PRINT "      SALESPERSON REIMBURSEMENT"
25 PRINT:PRINT
30 PRINT "SALESPERSON      MILES      TOTAL"
40 PRINT " NAME          TRAVELED  REIMBURSEMENT"
45 PRINT
50 IS = "\      \      ####      #,###.##"
100 READ N$, M
110 IF N$ = "EOD" THEN 200
120 IF M < 1000 THEN R = M * .1
130 IF M >= 1000 AND M < 2000 THEN R = M * .12
140 IF M >= 2000 AND M < 3000 THEN R = M * .14
150 IF M > 3000 THEN R = M * .16
160 PRINT USING IS; N$, M, R
170 GOTO 100
180 DATA JERRY HILL, 2494, LOUSIE DAWES, 3786, WENDY JAMES, 695
190 DATA SHARON AIKEN, 1768, EOD, 0
200 END
```

**MODULE G: TOTALS**

**Part I** Evaluated to the satisfaction of the instructor.

**Parts II & III**

|                             |                                 |
|-----------------------------|---------------------------------|
| 30 REM N\$                  | NAME                            |
| 40 REM RES\$                | RESIDENCE                       |
| 50 REM HRS                  | HOURS                           |
| 60 REM TUI                  | TUITION                         |
| 70 REM TOTHR\$              | TOTAL HOURS                     |
| 80 REM TOTTUI               | TOTAL TUITION                   |
| 90 REM IN-STATE             | NUMBER OF IN-STATE STUDENTS     |
| 95 REM OUT-STATE            | NUMBER OF OUT-OF-STATE STUDENTS |
| 200 PRINT B "NAME           | RESIDENCE HOURS TUITION"        |
| 210 READ N\$, RES\$, HRS    |                                 |
| 220 IF N\$ = "END" THEN 400 |                                 |

**Parts II & III (Continued)**

```
230 IF RES$="I" THEN TUI = HRS*16: IN-STATE = IN-STATE+1
240 IF RES$="O" THEN TUI = HRS*100: OUTSTATE = OUTSTATE+1
250 TOTHR$ = TOTHR$+HRS
260 TOTUI = TOTUI+TUI
270 PRINT USING "\      \      \      \#####.##";N$,RES$,HRS,TUI
280 GOTO 210
400 PRINT
410 PRINT USING "TOTALS      #####$,###,##";TOTHR$,TOTTUI
420 PRINT
430 PRINT "NUMBER OF STUDENTS IN-STATE  "INSTATE
440 PRINT "NUMBER OF STUDENTS OUT OF STATE"OUTSTATE
1000 DATA JOHNSON,I,12,BANKS,I,16,HAGGERTY,O,12,POWELL,I,16,
      MCDANIEL,O,16,END,O,0
```

**MODULE H: MODULE PROGRAMMING**

**Part I**            Evaluated to the satisfaction of the instructor.

**Part II**

**Project 1**

Add the line of--      2040 RETURN

**Project 2**

```
120 GOSUB 3000: REM DOES CALCULATIONS FOR THE REPORT
130 GOSUB 2000: REM PRINT THE OUTPUT REPORT
```

**MODULE I: CONTROLLED LOOPS**

**Project 1**

```
30 REM C    CELSIUS
40 REM F    FAHRENHEIT
80 PRINT "TABLE OF CORRESPONDING TEMPERATURES"
90 FOR F = 0 TO 220
100 C = 5/9*(F-32)
110 PRINT F,C
120 NEXT F
130 END
```

455



## Project 2

Appropriate tests: Input a normal range of commissions, enter commissions outside an acceptable range, enter word when numbers are requested.

```
30 REM N$          NAME
40 REM RATE        RATE OF COMMISSION
50 REM SALES       AMOUNT OF SALES
60 REM GROSS       GROSS PAY
70 REM TOTGROSS    TOTAL GROSS PAY
80 REM TOTSALES    TOTAL SALES
90 REM NS          NUMBER OF SALESPeOPLE
100 CLS
110 INPUT "HOW MANY SALESPeOPLE WILL YOU ENTER?";NS
120 FOR X = 1 TO NS
130 INPUT "ENTER SALESPerson'S NAME"N$
140 INPUT "ENTER RATE OF COMMISSION"RATE
150 INPUT "ENTER AMOUNT OF SALES"SALES
160 GROSS = RATE * SALES
170 TOTGROSS = TOTGROSS+GROSS
180 TOTSALES = TOTSALES+SALES
190 CLS
200 PRINT "    RATE OF    AMOUNT OF    GROSS"
210 PRINT "NAME    COMMISSION    SALES    PAY"
220 PRINT USING"\    \###.##    ###.##
    ###.##";N$,RATE,SALES,GROSS
230 NEXT X
300 PRINT USING"\    \ $###.##$###.##"; "TOTALS",
    TOTSALES, TOTGROSS
```

NOTE: Please provide data and END statement.

## MODULE J: ADVANCED CONCEPTS

### Parts I-IV

Determine number of parts.  
Dimension variables.  
Establish loop.  
Read part number and price.  
Continue or exit loop.  
Establish loop.  
Print part number.  
Continue or exit loop.  
Input number of part desired.  
Establish loop.  
If part number entered = part number, then print part number, price.  
Continue loop.

Parts I-III (Continued)

|                                              |                           |
|----------------------------------------------|---------------------------|
| 40 REM N\$                                   | PART NUMBER               |
| 50 REM PR                                    | PRICE                     |
| 60 REM NI                                    | NUMBER OF INVENTORY ITEMS |
| 70 REM PF\$                                  | PART TO FIND              |
| 100 NI=5                                     |                           |
| 120 DIM PN\$(NI),PR(NI)                      |                           |
| 130 FOR X=1 TO NI                            |                           |
| 140 READ PN\$(X),PR(X)                       |                           |
| 150 NEXT X                                   |                           |
| 160 FOR X = 1 TO NI                          |                           |
| 170 PRINT PN\$(X)                            |                           |
| 180 NEXT X                                   |                           |
| 190 INPUT "ENTER PART NUMBER";PF\$           |                           |
| 200 FOR X = 1 TO NI                          |                           |
| 210 IF PF\$=PN\$(X) THEN PRINT PN\$(X),PR(X) |                           |
| 220 NEXT X                                   |                           |

NOTE: Please provide data and END statement.

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(3)(4)(5)

The student shall be provided opportunities to demonstrate the skills related to algorithmic approaches to problem solving, programming language, and program execution, testing, debugging, and documentation.

| OBJECTIVE: | Students will demonstrate the skills associated with a programming language and the skills related to program execution, testing, debugging, and documentation. | MASTERY | NON-MASTERY |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|-------------|
|            |                                                                                                                                                                 |         |             |

#### MODULE A: SETUP PROCEDURES

- |    |                                                                                      |       |       |
|----|--------------------------------------------------------------------------------------|-------|-------|
| 1. | Demonstrate the ability to power-up, load the compiler, and power-down the computer. | _____ | _____ |
| 2. | List the steps of problem solving.                                                   | _____ | _____ |

#### MODULE B: INTRODUCTION TO PROGRAMMING

- |    |                                                          |       |       |
|----|----------------------------------------------------------|-------|-------|
| 1. | Identify keywords and commands.                          | _____ | _____ |
| 2. | Code, key, save, load, and execute a program.            | _____ | _____ |
| 3. | Edit program lines.                                      | _____ | _____ |
| 4. | Write simple programs using the keywords REM, PRINT,END. | _____ | _____ |

#### MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING

- |    |                                           |       |       |
|----|-------------------------------------------|-------|-------|
| 1. | Explain the benefits of documentation.    | _____ | _____ |
| 2. | Identify the three major types of errors. | _____ | _____ |
| 3. | Develop test data for a program.          | _____ | _____ |

## COMPETENCY CHECKLIST (Continued)

|  | MASTERY | NON-<br>MASTERY |
|--|---------|-----------------|
|--|---------|-----------------|

### MODULE D: USING ARITHMETIC

- |    |                                       |       |       |
|----|---------------------------------------|-------|-------|
| 1. | Define the types of variables.        | _____ | _____ |
| 2. | Use arithmetic operators.             | _____ | _____ |
| 3. | Develop arithmetic equations.         | _____ | _____ |
| 4. | Write programs using the keyword LET. | _____ | _____ |

### MODULE E: INPUT CONCEPTS

- |    |                                                                                                    |       |       |
|----|----------------------------------------------------------------------------------------------------|-------|-------|
| 1. | Demonstrate the use of data terminators.                                                           | _____ | _____ |
| 2. | Write programs using the keywords READ/DATA, IF...THEN, GOTO, and INPUT.                           | _____ | _____ |
| 3. | Demonstrate an output report by using proper headings (major, minor, and column), and PRINT USING. | _____ | _____ |

### MODULE F: BRANCHING

- |    |                                                                                  |       |       |
|----|----------------------------------------------------------------------------------|-------|-------|
| 1. | Identify the relational operators.                                               | _____ | _____ |
| 2. | Identify the logical operators.                                                  | _____ | _____ |
| 3. | Describe the different between a conditional branch and an unconditional branch. | _____ | _____ |
| 4. | Write programs using the conditional branch.                                     | _____ | _____ |

## COMPETENCY CHECKLIST (Continued)

|  | MASTERY | NON-<br>MASTERY |
|--|---------|-----------------|
|--|---------|-----------------|

### MODULE G: TOTALS

- |    |                                                                  |       |       |
|----|------------------------------------------------------------------|-------|-------|
| 1. | Define totaling and counting.                                    | _____ | _____ |
| 2. | Demonstrate how subtotals are accumulated and printed.           | _____ | _____ |
| 3. | Write programs to demonstrate totaling, counting, and subtotals. | _____ | _____ |

### MODULE H: MODULE PROGRAMMING

- |    |                                                     |       |       |
|----|-----------------------------------------------------|-------|-------|
| 1. | Define modular programming.                         | _____ | _____ |
| 2. | Write programs to use a main module and submodules. | _____ | _____ |

### MODULE I: CONTROLLED LOOPS

- |    |                                             |       |       |
|----|---------------------------------------------|-------|-------|
| 1. | Write programs using the keywords FOR/NEXT. | _____ | _____ |
|----|---------------------------------------------|-------|-------|

### MODULE J: ADVANCED CONCEPTS

- |    |                                                   |       |       |
|----|---------------------------------------------------|-------|-------|
| 1. | Write programs using tables to store data.        | _____ | _____ |
| 2. | Write programs using sort routines.               | _____ | _____ |
| 3. | Write programs using advanced character handling. | _____ | _____ |
| 4. | Write programs using graphics.                    | _____ | _____ |

STUDENT NAME \_\_\_\_\_

### SECTION III: PROGRAM DEVELOPMENT

#### UNIT I: PROBLEM-SOLVING CONCEPTS AND APPLICATIONS

**UNIT  
OBJECTIVE:**

Students will demonstrate the skills related to algorithmic approaches to problem solving, and translate problems into logical steps using appropriate program design techniques.

**STUDENT**

**COMPETENCIES:** After completing this unit, the student will be able to:

1. Define structured programming and related terminology.
2. Identify three types of control structures.
3. Demonstrate problem-solving techniques.
4. Utilize flowcharting and top-down analysis.

**FOCUS:**

In order to be able to write structured programs in an acceptable time frame, students must be skilled in the development of problem-solving techniques, program design, and program documentation.

## TERMINOLOGY

|                        |                                                                                                                                                                                                                                                                                        |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ADA                    | Multipurpose language developed by the U.S. Department of Defense due to their needs for a standardized language. Named to honor Ada Lovelace. Most recent and sophisticated of procedure-oriented languages.                                                                          |
| Algorithm              | An algorithm is a set of step-by-step instructions to solve a problem. Once an algorithm has been established, the actual coding of the program is greatly simplified, no matter what language is being used.                                                                          |
| APL                    | A Programming Language--symbolic interactive language used primarily by engineers, mathematicians, and scientists. Provides programmers with a "shorthand" that speeds up the process of coding. It is, however, hard to read and requires large amounts of computer storage space.    |
| Assembly language      | An improvement over machine language, which uses symbols or mnemonics to code instructions. It is easier than machine language but still very time consuming and detailed.                                                                                                             |
| BASIC                  | Beginner's All-Purpose Symbolic Instruction Code. A general purpose language for beginners which is the primary language for microcomputers. Very easy to learn and flexible but slow and lacks standardization.                                                                       |
| C                      | AT&T developed a general, low-level language that would work on all computers. It is machine-independent and transportable, but yet it provides the control of an assembler. It is structured, allowing the programming to build complex programs out of simple parts.                 |
| Chief programmer teams | A chief programmer is responsible for the project and is the primary writer of code. Others on the team back up and support the chief programmer by writing code and locating needed documents.                                                                                        |
| COBOL                  | Common Business Oriented Language--was developed in 1960 to be used in business applications. It is a procedure-oriented language and is used heavily in commercial and government computer installations. COBOL is easy to read and transportable but requires a very large compiler. |
| Coding                 | Translating the algorithm developed into a programming language. The sequence or structure of the program will be determined by the language used, but it should always be logical and understandable.                                                                                 |



**BCPI/Program Development**  
**Unit 1: Problem Solving Concepts and Applications**  
**Student's Guide/TERMINOLOGY**

|                                  |                                                                                                                                                                                                      |
|----------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Compiler</b>                  | A compiler does the interpretation just once and transfers the high-level language into machine code. A compiled program runs much faster but cannot be changed or corrected after it is compiled.   |
| <b>Computer program</b>          | A set of instructions to perform a specific task.                                                                                                                                                    |
| <b>Connector symbol</b>          | A circle is used as a connector between the different parts of the flowchart. It indicates movement from one place to another in the flowchart.                                                      |
| <b>Controlled loops</b>          | Controlled loops are used to repeat an action. Controlled loops enable the programmer to cause an action to be repeated a certain number of times or until a certain condition exists.               |
| <b>Debugging</b>                 | Testing and correcting problems found in the program's coding. Debugging should begin before the program is entered into the computer.                                                               |
| <b>Decision symbol</b>           | A diamond is used to indicate a decision point. Program control goes one way or another based on whether a condition is true.                                                                        |
| <b>Desk-checking</b>             | A form of debugging; includes examination of program logic, program completeness, and manually tracing the flow of data through the program.                                                         |
| <b>Documentation</b>             | The written explanation of the program, which is important to the programmer as well as to the program user. Good documentation is important if there is ever a need to modify or improve a program. |
| <b>Egoless programming teams</b> | There is no designated chief, and assignments are determined in a democratic way with each member doing that part of the work for which he/she is best-suited.                                       |
| <b>External documentation</b>    | Written documentation that accompanies the program. The larger or more detailed the program, the more important it becomes. Several things should be included in the documentation.                  |
| <b>Flowchart</b>                 | A flowchart is a graphic representation to show the steps in a process. In a computer program, it is a pictorial representation of the steps to be performed to produce the needed output.           |
| <b>Flowchart symbols</b>         | Make flowcharts simple to read and understand. The lines and arrows indicate the flow of action. The normal flow is from top to bottom and from left to right.                                       |



|                                        |                                                                                                                                                                                                                                                                 |
|----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>FORTH</b>                           | A fairly low-level language which was developed for microcomputers in the 1970s. It provides excellent control over the computer because it is close to assembly language and is often used in industrial applications.                                         |
| <b>FORTRAN</b>                         | FORmula TRANslator--used primarily for solving scientific, engineering, and mathematical problems. Developed about 1956. It is the world's oldest high-level language. It is very popular but has problems with compatability.                                  |
| <b>GOTO</b>                            | Directs the program to the line number following the word GOTO. The fewer the number of GOTOs in a program the better and more structured the program.                                                                                                          |
| <b>Hierarchical grouping</b>           | A programming manager assigns the coding to an individual programmer without actually participating in the coding himself/herself.                                                                                                                              |
| <b>High-level languages</b>            | Languages that use English-like terms and commands to make programming and understanding programs written by others easier. In most applications they are easier to learn, require less time to write, are not machine-dependent, and are easier to understand. |
| <b>Immediate mode</b>                  | Requires that a language be loaded. However, line numbers are not needed. After each statement is typed, press the ENTER key and the statement will be executed.                                                                                                |
| <b>Incorrect or inappropriate data</b> | Should be input to test how the program deals with it. Does the program give incorrect results or stop running, or are there proper error handling capabilities?                                                                                                |
| <b>Input</b>                           | A method of assigning a value to a variable.                                                                                                                                                                                                                    |
| <b>Input/output symbol</b>             | A parallelogram is used to show actions that accept input or output.                                                                                                                                                                                            |
| <b>Internal documentation</b>          | Explanation of a program which is located within the program itself.                                                                                                                                                                                            |
| <b>Interpreter</b>                     | An interpreter takes each instruction written in a high-level language and translates that instruction into machine code.                                                                                                                                       |
| <b>IPO chart</b>                       | A chart that identifies the input needed, the processing involved, and the output required. A programmer must first understand the output desired, then determine the input required and what processing must be done.                                          |
| <b>Language translators</b>            | High-level languages must be translated into machine language in order to be processed by the computer. This function is achieved by the language translator.                                                                                                   |

**BCPI/Program Development**  
**Unit 1: Problem Solving Concepts and Applications**  
**Student's Guide/TERMINOLOGY**

|                                     |                                                                                                                                                                                                                                                                                                                                            |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Logic errors</b>                 | Errors in the basic logic of the program. Perhaps the algorithm was not correct, or the program is not appropriate for the algorithm.                                                                                                                                                                                                      |
| <b>Machine language</b>             | Inside the computer, instructions are processed electronically in the binary language of the computer, machine language. Machine code is very time-consuming and difficult to write.                                                                                                                                                       |
| <b>Micro/modular flowchart</b>      | Shows the processing steps within the specified module of a program.                                                                                                                                                                                                                                                                       |
| <b>Non-numeric/string variables</b> | Used to represent words, letters or combinations of letters and numbers (for example a street address).                                                                                                                                                                                                                                    |
| <b>Numeric variables</b>            | Used to represent a number.                                                                                                                                                                                                                                                                                                                |
| <b>PASCAL</b>                       | Named for Blaise Pascal, a French inventor. This language is probably experiencing the most rapid growth rate. Very powerful, flexible, and self-documenting. It requires less memory, is easier to learn, and executes faster than other languages.                                                                                       |
| <b>PL/1</b>                         | Programming Language 1--IBM's answer to language with shortcomings such as FORTRAN and COBOL. Can be used for both scientific and business applications. This is a problem-oriented language. It is a modular language that is extensive and sophisticated, but its compiler is very large, which has prevented the use of PL/1 on micros. |
| <b>Print statement</b>              | Causes output to be displayed to the screen or sent to the printer.                                                                                                                                                                                                                                                                        |
| <b>Process symbol</b>               | A rectangle is used to represent some action or calculation.                                                                                                                                                                                                                                                                               |
| <b>Program/macro flowchart</b>      | Shows the major tasks or modules to be performed.                                                                                                                                                                                                                                                                                          |
| <b>Pseudocode</b>                   | A set of instructions using phrases in English describing what action should take place. It allows the programmer to think through a problem but is less time-consuming than flowcharting.                                                                                                                                                 |
| <b>Read and data</b>                | Read and data permit information to be stored in the program. It is a method of assigning a value to a variable.                                                                                                                                                                                                                           |
| <b>REM</b>                          | Means remark. REMs are used to document a program.                                                                                                                                                                                                                                                                                         |
| <b>RPG</b>                          | Report Program Generator--developed in about 1964 as a special purpose language for processing data to generate and print reports.                                                                                                                                                                                                         |

|                                       |                                                                                                                                                                                                                                                                                                                                                                           |
|---------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Run-time error</b>                 | An error that causes program execution to stop prematurely.                                                                                                                                                                                                                                                                                                               |
| <b>Structured programming</b>         | A top-down programming technique in which programs are designed to avoid confusing intertwined or "spaghetti" programs. Complex programs are built upon a series of building blocks or modules.                                                                                                                                                                           |
| <b>Subscripted variables (tables)</b> | A table is a variable that can store more than one piece of data at a time. These are called subscripted variables. A table must be dimensioned to tell the program the largest number of elements that may be placed in the table.                                                                                                                                       |
| <b>Syntax</b>                         | Refers to the exacting way that commands must be typed in and arranged for the computer.                                                                                                                                                                                                                                                                                  |
| <b>Syntax errors</b>                  | Mistakes in the language being used. Often these are careless errors of keying in a word incorrectly.                                                                                                                                                                                                                                                                     |
| <b>System flowchart</b>               | Provides a broad overview of the processing operations that are to be accomplished by showing the data flow through all parts of the data processing system.                                                                                                                                                                                                              |
| <b>Terminal symbol</b>                | An oval is used to indicate the beginning and end of a flowchart.                                                                                                                                                                                                                                                                                                         |
| <b>Top-down design</b>                | Top-down analysis or design takes a problem and breaks it down into smaller parts; proceeds from the general to the specific.                                                                                                                                                                                                                                             |
| <b>Totaling and counting</b>          | Programs use an accumulator to add a series of numbers and produce a total. A variable is assigned as an accumulator; then each new amount is added to the number already in the accumulator. A counter is a variable that is incremented by one (or by whatever number you increment the counter). There can be conditional and unconditional accumulators and counters. |
| <b>Typical data</b>                   | Data that would typically be entered into the program.                                                                                                                                                                                                                                                                                                                    |
| <b>Unusual but valid data</b>         | It is good to have a range check to see the extremes the computer will handle.                                                                                                                                                                                                                                                                                            |
| <b>Variables</b>                      | Data is stored in memory locations called variables. The contents of the variables can change.                                                                                                                                                                                                                                                                            |

## UNIT OVERVIEW

- I. Structured programming
  - A. Structured programming is a logical style of design at the beginning of a project. A structured program is a program in which each block of code (or module) is made up on one of three logical control structures: sequence, selection, or looping.
  - B. Major contributions were made by Edsger Dijkstra, a Dutch computer engineer, in the 1960s.
  - C. A structured program is made up of segments, and each segment has only one entry point and one exit point.
  - D. Some advantages of structured programming are:
    - 1. The ease of checking proper logic.
    - 2. Debugging.
    - 3. Being able to change a program based on user needs.
- II. Control structures
  - A. Control structures are important to good programming logic.
    - 1. Three basic control structures (or patterns) are all that are needed to plan the most effective solution to any problem in structured programming.
    - 2. Control structures provide for proper sequence of steps for executing the program.
    - 3. If control structures are used properly, the program will be easy to follow, modify, and debug.
  - B. The three basic types of control structures are:
    - 1. **Sequence**--a structure pattern used to execute statements one after another in order of storage. After all statements have been executed, the problem is solved.
    - 2. **Selection**--structure used to compare data in a program. This pattern requires a test.
      - a. The comparison determines the action the program will take.

- b. The choices or decisions split a program to follow different paths.
- c. A logical test usually is the reason for this structure.
  - (1) If the test is proven to be true, one path of the program and logic flow is taken by the computer.
  - (2) If the test proves false, then another path is taken.
- 3. **Loop**--structure that tests the number of times the loop is executed.
  - a. Each time the loop is executed, the pass through is called an iteration.
  - b. There are two major types of loops.
    - (1) A **DOWHILE** loop continues to occur as long as the condition is **TRUE**. The test occurs immediately upon **ENTERING** the loop.
    - (2) A **DOUNTIL** loop test is at the **END** of the loop after all other processing within the loop has occurred. The basis of this loop is that the given condition is **FALSE**, and the loop continues to execute as long as that condition remains false.

### III. Solving the problem

#### A. Problem definition

- 1. The first and most important step in problem solving is making sure the problem is understood.
- 2. The problem needs to be defined in clear and precise terms that can be understood by everyone involved.
- 3. There must be a recognition that a problem or need exists before there can be a cooperative effort to define the problem.

#### B. Information desired

- 1. What information is needed?
- 2. The information should be in what form?
- 3. A detailed and well-planned description of the output should be determined.

C. Data required

1. What data are needed to provide the desired information?
2. Where can the data be found, and how will they be accumulated?
3. In what form and by what method should the data be entered?

D. Processing needed

1. How will the data have to be processed in order to produce the desired results?
2. What calculations will be required?
3. Is the programmer familiar with the algorithm necessary to do the processing, or will additional help or research be needed?

IV. **Designing the solution**--before a specific solution is determined several alternative plans might be examined. Creating and examining alternative ways to a solution often pin points flaws in some plans. Thinking through several alternatives in the beginning can prevent time-consuming trial and error later. At all times the programmer must keep in mind the output or end result desired. The solution must be able to be achieved in the acceptable time frame.

A. **Top-down design**--top-down analysis or design takes a problem and breaks it down into more manageable subprograms.

1. This is a design method which proceeds from the general to the specific.
2. First, all major functions are identified and then broken down into subfunctions.
3. This is a modular approach to programming in which a program is broken down into subunits.

B. **IPO chart**--a chart that identifies the input needed, the processing involved, and the output required. A programmer must first understand the output desired; then determine the input required and what processing must be done.

C. **Algorithm**--an algorithm is a set of step-by-step instructions to solve a problem.

1. Once a problem has been defined and understood, the process of solving the problem can begin.



2. Once an algorithm has been established, the actual coding of the program is greatly simplified, no matter what language is being used.
  3. Algorithms should be clear and precise and have a finite number of steps.
  4. Two popular ways of representing algorithms are with pseudocode and flowcharts.
- D. **Flowchart**--a flowchart is a graphic representation to show the steps in a process. In a computer program it is a pictorial representation of the steps to be performed to produce the needed output.
1. Types of flowcharts
    - a. **System flowchart**--provides a broad overview of the processing operations that are to be accomplished by showing the data flow through all parts of the data processing system.
    - b. **Program or macro flowchart**--shows the major tasks or modules to be performed.
    - c. **Micro or modular flowchart**--shows the processing steps within the specified module of a program.
  2. **Flowchart symbols**--make flowcharts simple to read and to understand. The lines and arrows indicate the flow of action. The normal flow is from top to bottom and from left to right.
    - a. **Terminal symbol**--an oval is used to indicate the beginning and end of a flowchart.
    - b. **Process symbol**--a rectangle is used to represent some action or calculation.
    - c. **Input/output symbol**--a parallelogram is used to show actions that accept input or output.
    - d. **Decision symbol**--a diamond is used to indicate a decision point. Program control goes one way or another based on whether a condition is true or false.
    - e. **Connector symbol**--a circle is used as a connector between the different parts of the flowchart. It indicates movement from one place to another in the flowchart.

3. Advantages of flowcharts
  - a. Makes coding of the program easier, faster, and more efficient.
  - b. Provides documentation of the program.
  - c. Illustrates any logic problems that might exist.
  - d. Helps in detecting, locating, and removing mistakes.
  - e. Makes modification to the program easier.
4. Disadvantages of flowcharts
  - a. Preparation of flowcharts is tedious and time-consuming.
  - b. Revisions in the program may require a completely new flowchart.
  - c. When programs involve several decisions, flowcharts are difficult to draw and understand.

E. **Pseudocode**--is a set of instructions using phrases in English describing what action should take place.

1. Pseudocode is a false code because pseudocode statements cannot be executed directly by the computer.
2. Advantages of pseudocode
  - a. Allows the programmer to think through a problem and concentrate on the steps required rather than the syntax of the language.
  - b. Less time-consuming than flowcharts.
  - c. Not tied to any particular language.
  - d. Easier to write than programming statements.
3. Disadvantages of pseudocode
  - a. Sometimes not as easy to understand as a graphic representation.
  - b. No standards or rules have been established for pseudocode.



## ACTIVITY 1: STRUCTURED PROGRAMMING

**INSTRUCTIONS:** Write a paragraph discussing the concepts of structured programming using as many terms as possible.

## ACTIVITY 2: THE THREE TYPES OF CONTROL STRUCTURES

INSTRUCTIONS: Draw the three types of control structures.

SEQUENCE

SELECTIONS

LOOP

### ACTIVITY 3: PROBLEM SOLVING

**INSTRUCTIONS:** Examine the following situations. In each situation define the problem, determine what information is desired, what data are required, and what processing will be needed.

#### Project 1

Lowlands' water supply seems to have a cash flow problem. It seems that their own bills are due each month before they receive payments from their customers.

Identify and define the problem:

Information desired:

Data required:

Processing needed:

### ACTIVITY 3 (Continued)

#### Project 2

Arnold's Electrical Supply never seems to have the right item when the customer comes in.  
On other items there seems to be an oversupply.

Identify and define the problem:

Information desired:

Data required:

Processing needed:

## ACTIVITY 4: TOP-DOWN ANALYSIS AND FLOWCHARTING

### Project 1

**INSTRUCTIONS:** Using top-down analysis explain how you would:

1. Determine your grade point average for one six weeks.
2. Determine the net income of a business for one year.



## ACTIVITY 4 (Continued)

### Project 3

**INSTRUCTIONS:** Draw the symbol for each of the following descriptors.

Start or End

Input or Output

Processing

Decision

Connection between parts of a flowchart

## ACTIVITY 4 (Continued)

### Project 4

**INSTRUCTIONS:** Draw a flowchart for the following problems.

1. Determine the square feet in a room that is 12 x 14.
2. Calculate the price on an item costing \$14.00 with a tax of 7%.
3. Calculate gross pay for an employee who is paid salary plus commission.
4. Determine the amount of a gas utility bill based on the meter readings of current and previous month.

1.

2.

3.

4.



## ACTIVITY 4 (Continued)

### Project 5

**INSTRUCTIONS:** Write pseudocode for the following problems:

1. Determine the amount of inventory at the end of the month.
  
  
  
  
  
  
  
  
  
  
2. Calculate the gross profit for an item of which 3 were sold.
  
  
  
  
  
  
  
  
  
  
3. Four people stayed two nights in the Sleepy Time Motel. At checkout time, the four teachers needed to split the bill. Write pseudocode to solve the problem.

## ACTIVITY 4 (Continued)

### Project 6

**INSTRUCTIONS:** Read each of the situations below and answer the questions.

1. A program has been written to keep student grades. The user of the program inputs the grades. An average for each student is displayed on the screen after the grades are entered. Answer the following question concerning testing of the program:

What data should be entered for testing?

2. A point-of-purchase program for an auto supply store has been written. When a customer makes a purchase, the part number and quantity are input into the computer. The computer produces a printout showing the items purchased, the quantity, the cost of each item, the tax, and the extension. The program provides totals which are shown on the printout. When the items are purchased, the program removes those items from the inventory count.

What would be appropriate tests for the program?

## **SECTION III: PROGRAM DEVELOPMENT**

### **UNIT II: PROGRAMMING CONCEPTS AND APPLICATIONS**

#### **UNIT**

#### **OBJECTIVE:**

Students will demonstrate the skills associated with a programming language and the skills related to program execution, testing, debugging, and documentation.

#### **STUDENT**

**COMPETENCIES:** After completing these modules, the student will be able to:

#### **MODULE A: SETUP PROCEDURES**

1. Demonstrate the ability to power-up, load the compiler, and power-down the computer.
2. List the steps of problem solving.

#### **MODULE B: INTRODUCTION TO PROGRAMMING**

1. Identify keywords and commands.
2. Code, key, save, load, and execute a program.
3. Edit program lines.
4. Write simple programs using the keywords REM, PRINT, END.

#### **MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING**

1. Explain the benefits of documentation.
2. Identify the three major types of errors.
3. Develop test data for a program.

#### MODULE D: USING ARITHMETIC

1. Define the types of variables.
2. Use arithmetic operators.
3. Develop arithmetic equations.
4. Write programs using the keyword LET.

#### MODULE E: INPUT CONCEPTS

1. Demonstrate the use of data terminators.
2. Write programs using the keywords READ/DATA, IF...THEN, GOTO, and INPUT.
3. Demonstrate an output report by using proper headings (major, minor, and column), and PRINT USING.

#### MODULE F: BRANCHING

1. Identify the relational operators.
2. Identify the logical operators.
3. Describe the difference between a conditional branch and an unconditional branch.
4. Write programs using the conditional branch.

#### MODULE G: TOTALS

1. Define totaling and counting.
2. Demonstrate how subtotals are accumulated and printed.
3. Write programs to demonstrate totaling, counting, and subtotals.

#### MODULE H: MODULE PROGRAMMING

1. Define modular programming.
2. Write programs to use a main module and submodules.

#### MODULE I: CONTROLLED LOOPS

1. Write programs using the keywords FOR/NEXT.

#### MODULE J: ADVANCED CONCEPTS

1. Write programs using tables to store data.
2. Write programs using sort routines.
3. Write programs using advanced character handling.
4. Write programs using graphics.

#### FOCUS:

Each programming language has its own set of rules, structure, and syntax. When students understand the structure and syntax of a language and develop program design skills, they are able to use a high-level language to solve problems and program a computer.

## OVERVIEW

### MODULE A: SETUP PROCEDURES

- I. **Powering-up, loading the compiler, and powering-down**--this will vary with the type of computer used.
- II. Studying programming can cause a great deal of frustration for the following reasons:
  - A. Concepts will have to be developed for successful problem solving and programming.
  - B. There may be more than one way to arrive at the correct output. Everyone's logic is not the same.
  - C. Working on a programming assignment requires a different approach from techniques that may be used in other subject areas.
    1. Programming is not like doing a set of review questions in history.
    2. Writing programs requires a lot of time.
    3. Several revisions, runs, and reruns must be made before the final version is completed.
    4. A programming assignment cannot be left until the last moment to complete.
    5. The solution must be well planned before the program is entered into the computer.
    6. Guidelines to successful programming.
      - a. Start early.
      - b. Allow for computer time.
      - c. Be patient.
      - d. Handle frustration.
      - e. Work hard to succeed.
  - D. Read and study your material for new concepts.
    1. If you do not understand the concept, please request assistance.

2. Do not fall into the trap of copying a friend's program.
3. Programming is like building a house, one step at a time.
4. Each level depends on the previous one.
5. It would be impossible to put the roof on a house if the walls were not firmly in place; problem solving through programming follows the same building concept.

E. **Programming process**--review the steps for solving the problem.

1. **Problem definition**--the first and most important step in problem solving is understanding what the problem is.
2. **Information desired**--decide what information is needed and the desired form of output.
3. **Data required**--determine what data are needed to provide the desired information, how they will be accumulated, and what form and by what method the data should be entered.
4. **Processing needed**--resolve how to process the data to produce the desired results; know what calculations will be required and whether the programmer is familiar with the algorithm necessary to do the processing or will need additional help or research.

## MODULE A: SETUP PROCEDURES

### ACTIVITY 1: POWER-UP, LOAD THE COMPILER, AND POWER-DOWN THE COMPUTER

#### Project 1

**INSTRUCTIONS:** Turn on the computer. Format and put an operating system on your disk according to the procedures demonstrated by your instructor.

#### Project 2

**INSTRUCTIONS:** Enter your name on the screen, and then enter a series of numbers that is provided by your instructor.



## ACTIVITY 2: PROBLEM SOLVING

**INSTRUCTIONS:** List and describe the four steps for solving a problem.

1.

2.

3.

4.

## OVERVIEW

### MODULE B: INTRODUCTION TO PROGRAMMING

- I. Using a high-level language--BASIC will be the language used in the following modules as examples.
  - A. **Semantics and syntax**--a computer is a very literal machine. It does exactly what it is programmed to do.
    1. High-level languages can interpret only the statements or keywords (reserved words) of the specific language. If these words are used incorrectly or misspelled, an error message will appear.
    2. **BASIC**--one of many programming languages.
      - a. Beginner's All-Purpose Symbolic Instruction Code (BASIC) was developed by Thomas Kurtz and John Kemeny of Dartmouth College in 1965.
      - b. BASIC has gained popularity and has evolved into a powerful language used for both business and scientific applications.
      - c. BASIC is supported from microcomputers to large, time-sharing systems. It is supported on more computers than any other language.
      - d. ANSI Standard BASIC is altered by hardware and software vendors to better accommodate their products; therefore, several slightly different versions of BASIC are on the market.
    3. Languages such as BASIC require line numbers and the ENTER key to be pressed at the completion of each line. Other languages may not require line numbers and/or use other conventions to terminate a line.
  - B. Many of the activities can be used with any high-level language.
  - C. The following are keywords and commands that are common to BASIC.
    1. **Immediate mode**--requires that a language be loaded; however, line numbers are not needed. After each statement is typed, press the ENTER key, and the statement will be executed.
    2. **REM**--means remark. REM is used to document a program. REM can be used to place the following information in the program:
      - a. Programmer's name
      - b. Name of program

- c. Date written
  - d. Variable names and purpose or type
  - e. Explanation of parts of a program
3. **PRINT**--is a keyword which causes printing to appear on the CRT screen. If a printer is being used, **LPRINT** will cause the printer to produce output.
- a. **Literals**--specific text that the computer can be instructed to print. A literal is anything enclosed in quotation marks. The **PRINT** statement will print any character that has been placed between the quotation marks. Examples would include the following:
    - (1) "HELLO"
    - (2) "TEST 1"
    - (3) "PROFIT/LOSS"
    - (4) "1st Semester"
  - b. **Constants**--actual numbers; other designated items that may be produced with the keyword **PRINT**. A constant is not placed inside quotation marks. Examples would include the following:
    - (1) 86
    - (2) 43.5
    - (3) 32
    - (4) 112.95
  - c. **Spacing**--can be controlled with commas and semicolons.
    - (1) **Commas**--are used to separate the items printed into zones. The number and size of the zones vary with the computer being used. Using commas is one method to print items in columns.
    - (2) **Semicolons**--are used to separate output information with no space inserted between the items printed.
4. **END**--is a keyword which causes the computer to stop executing a program. The **END** statement is optional when using most microcomputers, but it is a good habit to include this keyword in the program.

5. To communicate with the computer, system commands are used like program instructions. Systems commands are not part of the program. They are used to store, retrieve, and execute a program. The following commands are common to most computers.
  - a. **NEW**--clears primary storage. This statement should always be used before starting a new program.
  - b. **SAVE**--causes the program currently in memory to be saved on secondary storage, probably a disk.
  - c. **LOAD**--causes the program identified to be retrieved from secondary storage and loaded into memory.
  - d. **LIST**--causes all or part of the program currently in memory to be displayed on the CRT screen.
  - e. **LLIST**--causes the printer to print all or part of the program currently in memory.
  - f. **RUN**--causes the program in memory to be executed.

II. **Coding the program**--involves translating the algorithm developed within a programming language. The sequence or structure of the program will be determined by the language used, but it should always be logical and understandable. Careful planning is essential in making the program logical and ensuring that the program will be completed in the desired time.

- A. **Programmer organization**--a program was once considered the product of an individual working alone. This still can be the case, but more often it involves several people working together. Programmers today must be able to work in this type of programming environment.
  1. **Hierarchical grouping**--a programming manager assigns the coding to another individual and does not actually participate in the coding task(s).
  2. **Chief programmer teams**--a chief programmer is responsible for the project and is the primary writer of code. Others on the team back up and support the chief programmer by writing code and locating needed documents.
  3. **"Egoless" programming teams**--there is no designated chief, and assignments are determined in a democratic way with each member doing the part of the work for which he/she is best suited.
- B. **Structure**--the structure of computer programs should be logical, orderly, and efficient.

1. Programs using a structured technique are easier to understand, faster to code, and easier to debug. If a program is not properly structured and clearly written, no one else can understand it; sometimes even the original programmer has difficulty.
  2. Programs should logically flow from one part to another and be easy to follow. Intertwined or "spaghetti" programs should be avoided.
  3. Care must be taken to allow only one entrance into and one exit from a structure. Minimize the use of branching instructions.
  4. Complex programs are built upon a series of building blocks or modules.
- III. **Editing**--changing the program is accomplished by adding data, deleting data, or modifying current lines.
- A. Add a line by simply keying in the line number, followed by the program statement.
  - B. Delete a single line by keying in the line number and pressing the ENTER or RETURN key.
  - C. Modify a line with the edit command, or reenter the line number and the new program statement.

## MODULE B: INTRODUCTION TO PROGRAMMING

### ACTIVITY 1: IDENTIFYING KEYWORDS AND COMMANDS

**INSTRUCTIONS:** Identify the following items by placing the correct letter in the blank.

A. Keyword                      B. Command

- |       |     |        |
|-------|-----|--------|
| _____ | 1.  | REM    |
| _____ | 2.  | PRINT  |
| _____ | 3.  | END    |
| _____ | 4.  | SAVE   |
| _____ | 5.  | LOAD   |
| _____ | 6.  | LLIST  |
| _____ | 7.  | NEW    |
| _____ | 8.  | RUN    |
| _____ | 9.  | LPRINT |
| _____ | 10. | LIST   |

## ACTIVITY 2: BASIC PROGRAMMING TECHNIQUES

**INSTRUCTIONS:** Key in each program and save to a disk. Use your first name, and number your programs 1, 2, 3 (i.e., John1, John2, John3). Make sure to use NEW or whatever keyword is appropriate for your hardware to clear the computer memory of the previous program before the next one is entered.

### Program 1

```
10 PRINT "MY NAME IS JERRY"  
20 PRINT "THE PERSON NEXT TO ME IS NANCY"
```

### Program 2

```
40 PRINT "5 + 5 + 10"  
50 PRINT "IS" 5 + 5 + 10
```

### Program 3

```
10 PRINT "THE GRADES FOR ALICE ARE "  
20 PRINT "90 80 70 86"  
30 PRINT "THE AVERAGE FOR ALICE IS ";  
40 PRINT (90+80+70+86)/3
```

### ACTIVITY 3: EDITING PROGRAM LINES

#### Project 1

**INSTRUCTIONS:** Load program 1 from Activity 2. Modify the program using your own name and the name of the person sitting next to you. Screen print the listing of the program; then run the program, and screen print the output. Save as program 4 (i.e., John4).

#### Project 2

**INSTRUCTIONS:** Load program 2 from Activity 2. Modify it by changing the numbers as directed by your teacher. Save as program 5 (i.e., John5).

#### Project 3

**INSTRUCTIONS:** Load program 3 from Activity 2. Modify it by changing the name to your name and the numbers according to your teacher's directions. In line 40, replace the /3 with /4. Save as program 6 (i.e., John6).



## ACTIVITY 4: WRITING SIMPLE PROGRAMS

### Project 1

INSTRUCTIONS: What is the output of the following program segments?

1.     10 PRINT "HI! "  
       20 PRINT "WELCOME TO THE WORLD OF PROGRAMMING."
  
2.     10 PRINT "HI! ";  
       20 PRINT "WELCOME TO THE WORLD OF PROGRAMMING."
  
3.     10 PRINT "THE ", "END"
  
4.     10 PRINT "THE "; "END"
  
5.     10 PRINT "THE ";  
       20 PRINT "END"

## ACTIVITY 4 (Continued)

### Project 2

**INSTRUCTIONS** Write a program to produce the following output:

THE NAME OF MY COMPUTER TEACHER  
IS  
MR. JEFFERSON

### Project 3

**INSTRUCTIONS:** Modify the above program to print the correct name of your computer teacher.

### Project 4

**INSTRUCTIONS:** Write a program containing two print statements that produce the following output:

THIS WILL APPEAR ON ONE LINE

### Project 5

**INSTRUCTIONS:** Code a program to produce the following output:

- a. NOTICE THE EFFECTS OF COMMAS
- b. YOU CAN ALSO  
SKIP LINES

## OVERVIEW

### MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING

- I. **Documentation**--is an explanation of the program and is important to the programmer as well as to the program user. Good documentation is important if there is ever a need to modify or improve a program.

A. Benefits of documentation include:

1. Firm control over resources by providing a system of checks and balances.
2. Possible reduction in operating costs through more effective utilization of resources.
3. Means of measuring the effectiveness of operation, or organization, by applying a realistic performance standard.
4. Simplification of the process of training personnel.
5. Quality and accuracy in operations and user organization by clearly defining what output and/or results are desired.
6. Elimination of dependence upon individuals in an operation or organization.

- B. **Internal documentation**--is located within the program itself; thus, there is no separate book or manual to locate. Internal comments are used to make the program easier for people to understand. Comments mean nothing to the computer.

1. In BASIC, internal documentation is made with REM statements.
2. State the purpose of the program as a whole.
3. Use variable names that give an indication of what the variable represents.
4. Variables should be identified at the beginning of the program, followed by a brief explanation of each variable's purpose.
5. Comments describing each module and how it is processed should be clearly marked.
6. Comment boxes can be used as labels.
7. Comments should document program changes.

- C. **External documentation**--is the written explanation that accompanies the program. The larger or more detailed the program, the more important external documentation becomes. Several things should be included in the documentation package.
1. **Narrative description**--is a description of what the program does, why it does it, for whom it is intended, and what language is used.
  2. **Frequency chart**--indicates how often the program is run. (Is it run daily, weekly, or monthly?)
  3. **Design documents**--include hierarchy charts, IPO chart, flowcharts, and/or pseudocode.
  4. **Variable list**--contains the name and description of each variable used in the program.
  5. **Input/output layouts**--are the screen layout chart and the printer spacing chart.
  6. **Source listing**--has embedded (internal) documentation comments.
  7. **Operator instructions**--state starting, running, and terminating procedures.
  8. **Test data**--are used to test and debug the program after each program change.
  9. **Correct output results**--are examples of the hardcopy reports.
  10. **Audit trail**--provides the date of and reason for program maintenance.
- D. **Company guidelines**--set the standard for a particular company; therefore, the documentation will vary with each company. Each company will determine what will be in the documentation package and where to locate standard comments.
- II. **Debugging**--time spent in debugging and testing often equals or exceeds the time spent in program coding. Debugging should begin before the program is entered into the computer. "Desk-checking" includes examination of program logic and program completeness and manually tracing the flow of data through the program.
- A. **Errors**--there are three major types of errors.
1. **Run-time error**--is an error that causes program execution to stop prematurely--"crash."
  2. **Syntax errors**--are mistakes in the language being used, which are normally caught by the compiler.
    - a. Often these are careless errors of keying in a word incorrectly.

- b. Coding errors are usually pointed out by the computer with an error message.
  - c. The error message may not always clearly define the problem, but it points you in the direction of the error.
  - d. Some careful probing may be required to find the exact cause of the error.
- 3. **Logic errors**--are errors in the basic logic, or construction, of the program.
  - a. Perhaps the algorithm was not correct or the program is not appropriate for the algorithm.
  - b. Logic error will only be discovered when the program is executed with valid data.
  - c. Examples of logic errors would be:
    - (1) Programming a statement in the wrong order.
    - (2) An arithmetic expression not written correctly.
- B. Common errors to look for when debugging a program are:
  - 1. Wrong variable lengths, including decimal places.
  - 2. Input/output descriptions that do not match specifications.
  - 3. Variables not properly initialized.
  - 4. Wrong data set used for testing.
  - 5. Calculation miscoded.
  - 6. A misplaced or no WRITE statement.
  - 7. A misplaced or no READ statement.
  - 8. No data supplied.
  - 9. IF-THEN-ELSE coded incorrectly.
- C. Debugging aids include:
  - 1. Error messages because of an abnormal ending.
  - 2. Using a trace.

3. Displaying memory contents.
  4. Using dumps.
  5. Using algorithms and other documentation.
  6. Teacher assistance should be given only when the student has given sufficient effort to the problem.
- III. **Testing**--program auditing is completed after the program is compiled and all diagnostic errors are removed. Test data should be developed to find errors in the program. The test data should require the program to execute all the instructions within the program at least one time.
- A. The purpose of testing is to determine if the results produced are correct.
    1. Does the report/screen layout match specifications?
    2. Are calculations correct?
    3. Are all possible error conditions handled properly?
    4. Are all records processed?
  - B. The testing procedure involves using the program to process input test data that will produce known results.
    1. Code and test main logic with good data.
    2. Add complexity one step at a time, and retest.
    3. Add error-trapping procedures, code, and test.
    4. It is beneficial to have someone unfamiliar with the program to run the test.
    5. The programmer knows what is expected, what the prompts mean, and the acceptable range of data to be used as input.
  - C. Create data for testing
    1. **Typical data**--data that would typically be entered into the program; valid records.
    2. **Unusual but valid data**--it is good to have a range check to see the extremes the computer will handle.
    3. **Incorrect or inappropriate data**--should be input to test how the program deals with it. Does the program give incorrect results or stop running, or are there proper error-handling capabilities?

4. Valid and invalid test data should be created to check data validation concepts--range checks, alphabetic checks, numeric checks, and limit checks.

D. Interpret computer output

1. Heading and detail lines show correct information.
2. Spacing and spelling are correct.
3. Calculated results are correct.
4. IF-THEN-ELSE paths are correct.
5. Invalid records are handled properly.
6. Totals are correct.



## MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING

### ACTIVITY 1: PROGRAM DOCUMENTATION

#### Project 1

**INSTRUCTIONS:** Examine the two programs provided by the instructor. Answer the following questions concerning the programs. Write NONE if not present in the documentation.

#### **Program 1**

1. Name of the programmer:
2. Variable names and what they represent:
3. Name of the modules and the purpose of the module:

#### **Program 2**

1. Name of the programmer:
2. Variable names and what they represent:
3. Name of the modules and the purpose of the module:



## ACTIVITY 1 (Continued)

### Project 2

**INSTRUCTIONS:** Examine the written documentation provided by your instructor, and then answer the following questions.

1. What is the purpose of the software?
2. What are the capabilities and limitations?
3. How do you start the program running?
4. Give an example of an error message.

## ACTIVITY 2: DEBUGGING

**INSTRUCTIONS:** Debug the following programs.

### Program 1

```
10 REM ***** PROGRAMMER'S NAME "  
20 A = 5  
30 B = 3  
40 C$ = 7  
50 A + B + C = D  
60 PRINT "D"
```

### Program 2

```
10 REM ***** NAME *****  
20 N1 = 2  
30 2N = 4  
40 P = N1 * 2N  
50 PRINT "N1 * N2" = N1 * N2
```

### Program 3

```
50 GIRL = MARY  
60 BOYS$ = JIM  
70 PRINT A "LOVES B$"
```

### ACTIVITY 3: DEVELOPING TEST DATA

#### Project 1

**INSTRUCTIONS:** Design a record layout for the following input record description.

| NAME OF FIELD                   | LENGTH OF FIELD |
|---------------------------------|-----------------|
| Course number                   | 8               |
| Course title                    | 15              |
| Teacher                         | 15              |
| Current enrollment              | 2               |
| Semester taught 1=fall 2=spring | 1               |
| Period taught                   | 1               |

#### Project 2

**INSTRUCTIONS:** Using the input information from Project 1, design two different printer spacing charts. These charts will show the computer-generated report to the user.

## OVERVIEW

### MODULE D: USING ARITHMETIC

- I. **Variables**--data are stored in memory locations called variables. The contents of the variables can change.
  - A. Rules for variable names
    1. May contain a number but cannot start with a number.
    2. May not be the same as a keyword or a command.
    3. May not contain a special symbol (i.e., %, #, +, or -).
    4. Should give an indication of what the variable represents (example: AMT for amount).
  - B. Types of variables
    1. **Numeric**--used to represent a number.
    2. **Non-numeric or string**--used to represent words, letters, or combinations of letters and numbers (for example, a street address). String variables are written with a "\$" as the last character; for example, NAME\$.
- II. **Arithmetic**--computers are well suited for doing arithmetic. A hierarchy exists for the order of operations.
  - A. Hierarchy of operations
    1. Operations inside parentheses
    2. Exponentiation
    3. Multiplication and division
    4. Addition and subtraction
    5. Operations performed from left to right
  - B. Symbols used
    1. \*\* or ^ Exponentiation
    2. \* Multiplication
    3. / Division

- 4. + Addition
- 5. - Subtraction

III. **LET**--is the keyword used to assign a value to a variable.

- A. The value can be a number, a formula, or another variable.
- B. The receiving variable name must always be on the left side of the equal sign.
- C. LET is optional on most computers.

## MODULE D: USING ARITHMETIC

### ACTIVITY 1: TYPES OF VARIABLES

**INSTRUCTIONS:** Identify the following items by placing the correct letter in the blank.

- |    |          |    |          |
|----|----------|----|----------|
| A. | Literal  | C. | Variable |
| B. | Constant | D. | Invalid  |

- \_\_\_\_\_ 1. "NAME"
- \_\_\_\_\_ 2. AGE
- \_\_\_\_\_ 3. C%
- \_\_\_\_\_ 4. 5.25
- \_\_\_\_\_ 5. 75.00
- \_\_\_\_\_ 6. COST
- \_\_\_\_\_ 7. "PRICE"
- \_\_\_\_\_ 8. .15
- \_\_\_\_\_ 9. "TAX"
- \_\_\_\_\_ 10. ITEM#

## ACTIVITY 2: USING ARITHMETIC OPERATORS

### Project 1

**INSTRUCTIONS:** Give instructions to the computer in the immediate mode.

1. Add 10.50, 27.26, and 36.10.
2. Add 90, 80, 70, and 95. Divide the result by 4.
3. Multiply 260 times 10.
4. Find 5 1/2% of the sum of 9.10, 12.24, and 5.

## ACTIVITY 2 (Continued)

### Project 2

**INSTRUCTIONS:** Code a program to solve the following problems:

1. Add 100, 200, 5000, and 250.
2. Square the number 9 and 123 and print the results in different zones of the screen.
3. Add 5, 6, and 9 and divide the results by 4.
4. Find the tax on the sum of \$5.80, 40.90 and 3.27 if the tax rate is 5.25%.

### Project 3

**INSTRUCTIONS:** Code a program to calculate gross pay. Hollie Hillis works 8 hours on Monday, 7 hours on Tuesday, 6 hours on Wednesday, 4 hours on Thursday, and no hours on Friday. Her pay is \$5.20 per hour.



### ACTIVITY 3: ARITHMETIC EQUATIONS

**INSTRUCTIONS:** What is the value of A in the following expressions?

1.  $A = 100 + 200 * 2$
2.  $A = 40 * 3.95$
3.  $A = 88 + 90 + 87 / 3$
4.  $A = 90 * 100 - 50 / 2$
5.  $A = ((40 + 4) * (50 - 5)) / 2$

512

## ACTIVITY 4: PROGRAMMING WITH THE KEYWORD LET

### Project 1

**INSTRUCTIONS:** Write a basic statement for each of the following using the keyword LET.

1. Assign the value of 0 to the variable A.
2. Input the formula  $A=X/Y$ .

### Project 2

**INSTRUCTIONS:** Write a BASIC program using the keyword LET that will input the values of A, B, and C; add the numbers and divide by 3.

## OVERVIEW

### MODULE E: INPUT CONCEPTS

- I. **Data terminator**--is a data item which is used in connection with the IF...THEN statement to terminate the program.
- II. Input statements
  - A. **READ** and **DATA**--are keywords which permit information to be stored in the program. These keywords must always appear together, **READ** cannot be used without **DATA**.
    - 1. **READ** and **DATA** is a method of assigning a value to a variable.
    - 2. **DATA** can be placed anywhere in the program but should be consistent in all programs so that they can be easily located and modified or checked.
    - 3. The number of data items must correspond to the number of items in the read statement.
    - 4. Items in a **DATA** statement must be separated by commas.
    - 5. The variable names and the data types must match.
  - B. **INPUT**--is a keyword in which an interactive program allows input from the keyboard.
    - 1. Inputs are a method of assigning a value to a variable.
    - 2. Inputs should be properly prompted so the program user knows what to input. This can be done with a **PRINT** statement prior to the input or by causing the input to print a message.
    - 3. Be sure the type of variable in the input statement is consistent with the type of variable needed.
  - C. **GOTO**--is a keyword which sends the program control to another line to start execution there.
    - 1. The **GOTO** statement is used to create a loop in the program.
    - 2. The **GOTO** statement is an unconditional branch. This means no condition is to be considered before branching.
  - D. **IF...THEN**--are keywords which tell the computer to make a decision.

1. The line number or statement will be executed if the condition is true; otherwise, the next program statement following the IF...THEN will be executed.
  2. The IF...THEN statement is a conditional branch. The computer must make a decision of which path to take next.
  3. The relational operator = is used as a condition which allows the computer to compare one value to another.
- III. **Output**--input data and information which are processed into usable form and then displayed on the screen or printed by the printer.
- A. PRINT causes output to be displayed on the screen.
  - B. Commas between quotation marks cause output to be displayed in different zones.
  - C. Semicolons at the end of the line causes the next PRINT statement to be executed on the same line as the previous line.
  - D. PRINT is used to display major, minor, and column headings.
  - E. PRINT USING is used to edit and format printed output. The PRINT USING statement uses an image to show how the output data should look.

## MODULE E: INPUT CONCEPTS

### ACTIVITIES 1 & 2: READ/DATA, IF...THEN, GOTO, AND INPUT

#### Project 1

**INSTRUCTIONS:** What would be printed if you had the following data and format?

| <u>DATA</u>           | <u>FORMAT</u> | <u>RESULTS</u> |
|-----------------------|---------------|----------------|
| 1. 423.68             | ###.#         | _____          |
| 2. .75                | ##.##         | _____          |
| 3. 9.68               | ###.##        | _____          |
| 4. 45.753             | #,###.##      | _____          |
| 5. 120.90             | \$#,###.##    | _____          |
| 6. 78.78              | \$\$#,###.##  | _____          |
| 7. 247.95             | \$\$#,###.##  | _____          |
| 8. +518               | ###-          | _____          |
| 9. -734               | ###-          | _____          |
| 10. THIS IS TOO SHORT | \ \           | _____          |

## ACTIVITIES 1 & 2 (Continued)

### Project 2

**INSTRUCTIONS:** What is the value for each of the variables on the specific attempt?

1. 10 READ H, W

.  
.  
.

80 DATA 35, 4.50, 40, 4.75

1ST TIME H = \_\_\_\_\_ W = \_\_\_\_\_

2ND TIME H = \_\_\_\_\_ W = \_\_\_\_\_

2. 10 READ A, B, C

.  
.  
.

90 DATA 5, 9, 4, 7, 8, 1, 3, 6

1ST TIME A = \_\_\_\_\_ B = \_\_\_\_\_ C = \_\_\_\_\_

2ND TIME A = \_\_\_\_\_ B = \_\_\_\_\_ C = \_\_\_\_\_

3RD TIME A = \_\_\_\_\_ B = \_\_\_\_\_ C = \_\_\_\_\_

3. 10 READ N\$, A

.  
.  
.

80 DATA JESSICA, 35.62, NATHAN

90 DATA 32.45, SHAWN, 15.62

1ST TIME N\$ = \_\_\_\_\_ A = \_\_\_\_\_

2ND TIME N\$ = \_\_\_\_\_ A = \_\_\_\_\_

3RD TIME N\$ = \_\_\_\_\_ A = \_\_\_\_\_

## ACTIVITIES 1 & 2 (Continued)

### Project 2 (Continued)

4. 10 READ N,IS

.

70 DATA 459, NOTEBOOK

80 DATA 123, PENCIL

1ST TIME            N = \_\_\_\_\_ IS = \_\_\_\_\_

2ND TIME           N = \_\_\_\_\_ IS = \_\_\_\_\_

### Project 3

**INSTRUCTIONS:** The following programs contain syntax errors. Find the errors, and rewrite the lines of code to eliminate the errors.

1. 10 READ N\$, A  
20 IF N\$ = "EOD" THEN 80  
30 PRINT N\$, A  
40 GOTO 10  
50 DATA 49.00, TOM  
60 DATA 25.00, BOB  
70 DATA 0, EOD  
80 END

2. 10 READ N C\$  
20 IF N = 0 THEN 80  
30 PRINT C\$, N  
40 GOTO 10  
50 DATA 1000 JONES  
60 DATA 2000 SMITH  
70 DATA 0, EOD  
80 END

## ACTIVITIES 1 & 2 (Continued)

### Project 4

**INSTRUCTIONS:** The following programs contain logic errors. Find the errors and rewrite the lines of code to eliminate the errors.

1.   10 READ D\$, N  
      20 IF D\$ = "EOD" THEN 10  
      30 PRINT D\$, N  
      40 GOTO 10  
      50 DATA TOYS, 4000, HARDWARE, 5000  
      60 DATA VIDEO, 8000, EOD, 0  
      70 END
  
2.   10 READ I\$, P  
      20 IF I\$ = "EOD" THEN 70  
      30 GOTO 10  
      40 PRINT I\$, P  
      50 DATA RULER, .25, PENCILS, .50  
      60 DATA RED PENS, .35, EOD, 0  
      70 END



## ACTIVITIES 1 & 2 (Continued)

### Project 5

**INSTRUCTIONS:** The following programs contain syntax errors. Find the errors and rewrite the lines of code to eliminate the errors.

1.     10 READ N\$, REGT, OVT  
       20 IF N\$ = "EOD" THEN 100  
       30 PAY = REGT + OVT  
       40 PRINT N\$, REGT, OVT, PAY  
       50 GOTO 10  
       60 DATA JERRY TICE, 250.00, 50.00  
       70 DATA DON MALOAN, 275.00, 15.00  
       80 DATA JOANN TAYLOR, 345.00, 25.00  
       90 DATA NO MORE, 0, 0  
       100 END
  
2.     10 READ I\$, SELLING, COST  
       20 IF I\$ = "EOD" THEN 100  
       30 PROFIT = SELLING - COST  
       40 PRINT I\$, SELLING, COST, PROFIT  
       50 GOTO 10  
       60 DESK 450.00, 345.00  
       70 CHAIR, 125.00, 75.00  
       80 LAMP, 75.00, 58.00  
       90 EOD, 0, 0  
       100 END

## ACTIVITIES 1 & 2 (Continued)

### Project 6

**INSTRUCTIONS:** The following programs contain logic errors. Find the errors and rewrite the lines of code to eliminate the errors.

1.   10 READ L, W  
      20 IF L = 0 THEN 70  
      30 PRINT L, W, A  
      40 A = L \* W  
      50 GOTO 10  
      60 DATA 10, 5, 14, 12, 0, 0  
      70 END
  
2.   10 READ S, CR  
      20 IF S = 0 THEN 70  
      30 CA = S \* CR  
      40 PRINT S, CR, CA  
      50 GOTO 20  
      60 DATA 4500.00, .18, 2800.00, .15, 0, 0  
      70 END

### ACTIVITY 3: OUTPUT

#### Project 1

**INSTRUCTIONS:** Enter and run the following program.

1. 

```
50 CLS
60 PRINT " WHAT IS YOUR NAME "
70 INPUT N$
80 PRINT "NAME A COLOR ";
90 INPUT C$
100 INPUT " NAME AN ANIMAL "A$
120 PRINT N$
130 PRINT "IS A "C$" "A$
```

Notice the different methods of combining prompts with inputs. Also notice how spaces are used in line 130 to keep the words from running together.

2. Make up your own program. The program should begin with a series of inputs, and then the variables input should be placed in a story. Debug the program so that no words run together.

#### Project 2

**INSTRUCTIONS:** Develop a thank-you letter with inputs to fill in the blanks. The output might appear as:

1. Dear xxxxxxxx,

Thank you so much for the xxxxxx. It was very nice of you. I received it xxxxxxxx. I am looking forward to seeing you xxxxxxxx.

Sincerely,

Your name

The location marked with X's should be input from the keyboard.

2. Write a math drill program. The program should:

Ask for the first number to be input.  
Ask for the second number to be input.  
Ask what the two numbers are when multiplied together.  
Ask for the answer to be input.  
Print the correct answer on the screen.

Flowchart the program before coding the program.

### ACTIVITY 3 (Continued)

#### Project 3

**INSTRUCTIONS:** The following programs contain syntax errors. Find the errors and rewrite the lines of code to eliminate the errors.

1. 

```
10 PRINT "ENTER STUDENT NAME: ";INPUT S$
20 PRINT "ENTER TEST SCORE 1: ";INPUT TEST1
30 PRINT "ENTER TEST SCORE 2: ";INPUT TEST2
40 AVG = (TEST1 + TEST2) / 2
50 PRINT "THE AVERAGE FOR "; S$; "IS "; AVG
60 PRINT "DO YOU HAVE MORE STUDENTS (Y/N) ";INPUT R$
70 IF R$ = "Y" THEN 10
80 IF R$ = "N" THEN 110
90 PRINT "PLEASE ENTER Y OR N "
100 GOTO 60
110 END
```
2. 

```
10 READ AMT, RATE
20 IF AMT = 0 THEN 70
30 DIS = AMT * RATE
40 PRINT AMT, RATE, DIS
50 GOTO 10
60 DATA 25.00, 25%, 49.00, 18%, 0, 0%
70 END
```

### ACTIVITY 3 (Continued)

#### Project 4

**INSTRUCTIONS:** The following programs contain logic errors. Find the errors, and rewrite the lines of code to eliminate the errors.

1.   10 READ N\$, H, R  
      20 IF N\$ = "EOD" THEN 90  
      30 PAY = H \* T  
      40 PRINT N\$, H, R, PAY  
      50 GOTO 10  
      60 DATA SUSAN WEST, 40, 6.50  
      70 DATA ROBERT MAPLE, 38, 7.00  
      80 DATA EOD, 0, 0  
      90 END
  
2.   10 READ C\$, OB, PMT  
      20 IF OB = 0 THEN 90  
      30 NB = OB - PMT  
      40 PRINT C\$, OB, PMT, NB  
      50 GOTO 90  
      60 DATA NANCY AUSTIN, 1800.00, 250.00  
      70 DATA DON MAUK, 2800.00, 300.00  
      80 DATA EOD, 0, 0  
      90 END

## MODULE E

### PROGRAM 1

**INSTRUCTIONS:** Design, code, key, and execute a program to produce a Salesperson List. Use the following data statements as input.

DATA HARDWARE, 349, STEVE WILLS

DATA SPORTING GOODS, 153, JERRY HILL

DATA SHOES, 279, TAMMY MOORE

DATA TOYS, 480, WENDY DAVIS

DATA VIDEO, 784, PAM ANDERSON

DATA EOD, 999, EOD

The report will be a three-column report containing the Salesperson Number, the Salesperson Name, and the Department. Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## MODULE E (Continued)

### PROGRAM 2

**INSTRUCTIONS:** Design, code, key, and execute a program to prepare a Sales Report. Use the following data as input.

| SALESPERSON<br>NAME | YTD<br>SALES | CURRENT<br>SALES |
|---------------------|--------------|------------------|
| EVEY HAYES          | 4500.00      | 1000.00          |
| SUSAN MAPLE         | 6700.00      | 2500.00          |
| ROBERT BATES        | 5495.00      | 2750.00          |
| NANCY AUSTIN        | 8945.00      | 1656.00          |
| DON WEST            | 3456.00      | 2340.00          |

The output will be a four-column report that will print the Salesperson Name, the YTD Sales, Current Sales, and Total Sales. Total Sales are to be calculated by adding YTD Sales and Current Sales. Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## MODULE E (Continued)

### PROGRAM 3

**INSTRUCTIONS:** Modify the Sales Report program to reflect a new total sales. Use the following data as input.

| SALESPERSON<br>NAME | YTD<br>SALES | CURRENT<br>SALES | SALES<br>RETURNS |
|---------------------|--------------|------------------|------------------|
| EVEY HAYES          | 4500.00      | 1000.00          | 575.00           |
| SUSAN MAPLE         | 6700.00      | 2500.00          | 1010.00          |
| ROBERT BATES        | 5495.00      | 2750.00          | 1750.00          |
| NANCY AUSTIN        | 8945.00      | 1656.00          | 56.00            |
| DON WEST            | 3456.00      | 2340.00          | 325.00           |

The output will be a five-column report that will print the Salesperson Name, the YTD Sales, Current Sales, Sales Return, and Total Sales. Total Sales are to be calculated by adding YTD Sales and Current Sales and then subtracting the Sales Returns. Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.



## MODULE E (Continued)

## PROGRAM 4

**INSTRUCTION:** Design, code, key, and execute a program to prepare a Sales Commission Report. The data will be entered into the program with the INPUT statement. Use proper prompts. Use the following as data.

| SALESPERSON<br>NAME | CURRENT<br>SALES | COMMISSION<br>RATE |
|---------------------|------------------|--------------------|
| SCOTT COLIER        | 4290.95          | 6%                 |
| MARY ORWAY          | 3578.78          | 10%                |
| RANDY TICE          | 6041.69          | 14%                |
| NINA MAUK           | 9752.34          | 15%                |
| JOHN DAVIS          | 8674.21          | 8%                 |

The output will be a four-column report including the Salesperson Name, Current Sales, Commission Rate, and Commission Amount. Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## OVERVIEW

### MODULE F: BRANCHING

- I. **Relational operators**--are used in BASIC to compare one value with another. The six relational operators are:
  - A. = Equal to
  - B. < Less than
  - C. > Greater than
  - D. <> Not equal to
  - E. <= Less than or equal to
  - F. >= Greater than or equal to
- II. **Logical operators**--are used in some versions of BASIC in the IF... THEN statement. The logical operators are:
  - A. AND
  - B. OR
  - C. NOT
- III. **Branching**--the course of action by the computer can be alternated between either conditional or unconditional.
  - A. The GOTO causes an unconditional branch in the sequence of execution.
    1. GOTO directs the program to the line number following the word GOTO.
    2. The fewer the number of GOTOs in a program, the better and more structured the program.
    3. Writing "GOTO-less" programs should be attempted.
  - B. The IF...THEN statement is a conditional branch.
    1. If a condition is evaluated as true, then control of execution can be transferred to some other place in the program, or the instruction following THEN is executed.
    2. If the condition is evaluated as false, then the next program statement is executed.

- C. **Looping**--controlled loops are used to repeat an action. Controlled loops enable the programmer to cause an action to be repeated a certain number of times or until a certain condition exists.
- D. **Data validation**--is referred to as input error-trapping.
  - 1. This process is used to check the input data, and if the data are not valid, then it requires the user to reenter the data.
  - 2. Methods that can be used are:
    - a. Range checking.
    - b. Checking the input options.
    - c. Checking for consistency.

## MODULE F: BRANCHING

### ACTIVITY 1: RELATIONAL OPERATORS

#### Project 1

**INSTRUCTIONS:** List which relational operator is being used in the following IF...THEN statements.

1. IF AMT = 0 THEN 999 \_\_\_\_\_
2. IF DISCOUNT <= .25 THEN 500 \_\_\_\_\_
3. IF COST <> 0 THEN 200 \_\_\_\_\_
4. IF SALES > 500.00 THEN 500 \_\_\_\_\_
5. IF AGE < 21 THEN 50 \_\_\_\_\_
6. IF TAX >= .07 THEN 400 \_\_\_\_\_

#### Project 2

**INSTRUCTIONS:** Indicate what makes each statement below invalid.

1. IF A = "21" THEN COST = RATE + 250.00
2. IF T = 50.00 OR > 50.00 THEN 90
3. IF RATE > 10% THEN 200
4. IF A > 250.00 OR IF A < 50.00 THEN 100
5. IF C IS GREATER THEN 99.99 PRINT "COST IS TOO HIGH --- ERROR"
6. IF CHOICE = 1 GOTO 1000
7. IF P\$ < A THEN 250
8. IF N\$ = EOD THEN 400
9. IF R\$ = YES OR R\$ = NO THEN 460
10. IF AMT > \$50.00 THEN 50

## ACTIVITY 2: LOGICAL OPERATORS

**INSTRUCTIONS:** Answer the following questions.

1. What are the three logical operators?
  - 1)
  - 2)
  - 3)
2. How are the logical operators used in BASIC?

### ACTIVITY 3: CONDITIONAL AND UNCONDITIONAL BRANCHING

#### Project 1

**INSTRUCTIONS:** In your own words, briefly define conditional and unconditional branching.

Conditional branching

Unconditional branching

### ACTIVITY 3 (Continued)

#### Project 2

**INSTRUCTIONS:** Predict the outcome of the following programs:

1.   40 READ A,B  
      50 IF A<0 THEN 200  
      60 TOTA = TOTA + A  
      70 TOTB = TOTB + B  
      80 GOTO 40  
      200 PRINT TOTA  
      210 PRINT TOTB  
      1000 DATA 4,2,21,33,2,4,21,9,2,6,0,2,-1,0
  
2.   40 READ A  
      50 IF A<0 THEN 200  
      60 IF A>100 THEN OH = OH + 1  
      70 IF A<100 THEN UH = UH + 1  
      80 GOTO 40  
      200 PRINT OH" NUMBERS OVER 100"  
      210 PRINT UH"NUMBERS UNDER 100"  
      1000 DATA 101,200,50,60,102,40,30,-1
  
3.   50 READ A,S  
      60 IF A=999 THEN 200  
      70 C=C+1  
      80 IF A>B THEN D=D+1:TOTA = TOTA+A  
      90 GOTO 50  
      200 PRINT C,D, TOTA  
      1000 DATA 4,5,10,6,12,10,3,8,9,2,100,50,6,7,999,999

## MODULE F

### PROGRAM 1

**INSTRUCTIONS:** Design, code, key, and execute a program to prepare a Sales Quota Report. This report is to list all employees with sales over \$5,000.00. Use the following as data.

| SALESPERSON<br>NAME | CURRENT<br>SALES |
|---------------------|------------------|
| LYNN HOLMAN         | 5409.00          |
| SUSAN CUMMINGS      | 4989.75          |
| BARBARA FARIN       | 3458.34          |
| JEFF PAUL           | 5000.00          |
| WALTER ANDERSON     | 8794.98          |
| SONDRA HILL         | 2050.99          |
| LORI ROSEN          | 7945.24          |
| WES THOMAS          | 6885.49          |
| JERRY STOOPS        | 1495.95          |
| ALVIN BELL          | 2567.89          |

The output will be a two-column report containing the Salesperson Name and Current Sales. Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.



## MODULE F (Continued)

### PROGRAM 2

**INSTRUCTIONS:** Modify the Sales Quota Report to list the salespeople with sales over \$5,000.00 who and have been employed with the company more than eight months. Use the following as data.

| SALESPERSON<br>NAME | CURRENT<br>SALES | MONTHS<br>EMPLOYED |
|---------------------|------------------|--------------------|
| LYNN HOLMAN         | 5409.00          | 19                 |
| SUSAN CUMMINGS      | 4989.75          | 15                 |
| BARBARA FARIN       | 3458.34          | 9                  |
| JEFF PAUL           | 5000.00          | 5                  |
| WALTER ANDERSON     | 8794.98          | 24                 |
| SONDRA HILL         | 2050.99          | 13                 |
| LORI ROSEN          | 7945.24          | 20                 |
| WES THOMAS          | 6885.49          | 18                 |
| JERRY STOOPS        | 1495.95          | 10                 |
| ALVIN BELL          | 2567.89          | 3                  |

The output will be a four-column report containing the Salesperson Name, Current Sales, Months Employed, and Amount Below Quota. If the salesperson has been employed for eight months and does not have a current sales amount over \$5,000.00, calculate the amount below quota and print as a field in the report. Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## MODULE F (Continued)

### PROGRAM 3

**INSTRUCTIONS:** Design, code, key, and execute a program to list the employees and the classification of employment. Use the following as data.

| EMPLOYEE       | CODE |
|----------------|------|
| AVTEL, DEAN    | 1    |
| BAILEY, DONNA  | 4    |
| CRUZ, JOE      | 3    |
| EADS, AMANDA   | 2    |
| HILLIER, BECKY | 5    |
| KATZKY, TARA   | 1    |
| MCCORD, BRUCE  | 3    |
| PEEK, DAVID    | 2    |
| ROSS, DEBRA    | 4    |
| STONE, RAY     | 6    |

MANAGEMENT employees are identified by a code of 1.  
FULL-TIME employees are identified by a code of 2.  
SWING SHIFT employees are identified by a code of 3.  
PART-TIME employees are identified by a code of 4.

The output will be a two-column report listing the Employee and the Classification of employment. Convert the code into the proper classification. Also use an error trap to identify invalid employment code. If the code is invalid, print the employee name and the message \*\*\* INVALID EMPLOYMENT CODE \*\*\*. Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## OVERVIEW

### MODULE G: TOTALS

- I. **Totaling and counting**--are the most frequent ways to summarize data. Summarizing shows the final results of calculations.
  - A. **Totals**--add up a series of numbers and produce a total or a sum. A variable is assigned as an accumulator; then each new amount is added to the number already in the accumulator. Examples would look like the following statements:
    1.  $TOTALCOST = TOTALCOST + COST$
    2.  $TOTALPAY = TOTALPAY + PAY$
    3.  $TOTALPOINTS = TOTALPOINTS + POINTS$
  - B. **Counters**--are variables that are incremented by one (or by whatever number you choose to increment the counter). A count will show how many items have been processed. Examples would look like the following statements:
    1.  $TOTALSALES = TOTALSALES + 1$
    2.  $STUDENTS = STUDENTS + 1$
    3.  $COUNTER = COUNTER + 2$
  - C. There can be conditional and unconditional accumulators and counters.
    1. An unconditional total or counter is set in the loop and will be done each time the data are processed. For example:
      - a. Counter-- $BOOKS = BOOKS + 1$
      - b. Total-- $TOTALF = TOTALF + F$
    2. A conditional total or counter is updated only when a certain condition occurs. This can be done by using an IF...THEN statement. For example:
      - a. Counter:  
 $IF\ H\$ = M\ THEN\ TOTALM = TOTALM + 1$
      - b. Total:  
 $IF\ H\$ = L\ THEN\ TOTALD = TOTALD + 75.00$
- II. **Subtotals**--are intermediate totals. At times, totals are required to be printed at intermediate points during the processing.

- A. A control variable is used as the key to determine when a subtotal is to be printed.
- B. A compare variable is used to determine when the control variable has changed.
- C. To process a subtotal, the control variable is compared with a compare variable (the previous record variable).
  - 1. When the variables are the same, processing continues.
  - 2. When the variables are different, a control break occurs.
    - a. The subtotal is printed, the subtotal variable is set back to zero, and the new control value is moved to the compare variable.
    - b. This continues until the last record has been read and processed.

## MODULE G: TOTALS

### ACTIVITY 1: TOTALING AND COUNTING

**INSTRUCTIONS:** Define totaling and counting.

Totaling

Counting

## MODULE G

### PROGRAM 1

**INSTRUCTIONS:** Modify the first Sales Quota Report in Module F to accumulate the following:

Total Salesperson  
Total Current Sales

These totals will reflect only the total salespersons over quota and that total of current sales.

Be sure to use proper headings for the total lines.

Use the proper documentation methods requested by the instructor.

### PROGRAM 2

**INSTRUCTIONS:** Modify the second Sales Quota Report in Module F to accumulate the following:

Total Salesperson  
Total Salesperson Below Quota

Be sure to use proper headings for the total lines.

Use the proper documentation methods requested by the instructor.

## MODULE G (Continued)

### PROGRAM 3

**INSTRUCTIONS:** Design, code, key, and execute a program to list an Accounts Receivable Register. The input consists of the Customer Number, Customer Name, and Invoice Amount. Use the following as data.

| CUSTOMER<br>NUMBER | CUSTOMER<br>NAME | INVOICE<br>AMOUNT |
|--------------------|------------------|-------------------|
| 4326               | CURRIE COMPUTERS | 450.87            |
| 4326               | CURRIE COMPUTERS | 54.00             |
| 4326               | CURRIE COMPUTERS | 35.00             |
| 7689               | JARRETT SOFTWARE | 234.65            |
| 7689               | JARRETT SOFTWARE | 87.00             |
| 7689               | JARRETT SOFTWARE | 123.90            |
| 7689               | JARRETT SOFTWARE | 67.83             |
| 1539               | WJ WAREHOUSE     | 342.00            |
| 1539               | WJ WAREHOUSE     | 653.00            |

The output is a detailed report showing each customer's total invoice amount. When there is a change in customer number, the accumulator will subtotal the amount. A final total is to be taken after all data have been processed. The detail line will contain all of the input fields.

Use the proper documentation methods requested by the instructor.

## OVERVIEW

### MODULE H: MODULE PROGRAMMING

- I. **Subroutines**--or modules are programs which are part of a larger program and perform a specific job.
  - A. For each function, a small program (or a submodule) is written. Submodules include:
    1. Programs to get the input data items.
    2. Programs to verify the input data.
    3. Programs to calculate/process necessary information.
    4. Programs to print the output.
  - B. A main module, or control module, is used to make the submodules operate in the proper order.
  - C. When planning submodules, the following should be kept in mind:
    1. One logical function will be performed in each module.
    2. The main module will call each module to do the proper task.
    3. Each submodule returns control to the main module when the specific task is completed.
    4. Each submodule will have only one way in and one way out.
- II. **GOSUB ... RETURN**--are the keywords used in modular programming.
  - A. **GOSUB**--is an abbreviation for GOto and SUBroutine. The GOSUB is used in the main module to transfer control to a submodule.
  - B. **RETURN**--is used at the end of the submodule to transfer control from the submodule back to the main module.



## MODULE H

### PROGRAM 1

**INSTRUCTIONS:** Design, code, key, and execute a program to find the gross pay for the Hill Printing Company. The input consists of the Employee, Hours Worked, and Rate. No employee should have over 40 hours. Use submodules in the program to get data, to check data, calculate, and print. Use the following as data.

| EMPLOYEE       | HOURS WORKED | RATE |
|----------------|--------------|------|
| CODY MACKIE    | 39           | 4.85 |
| ZACK CRAWFORD  | 40           | 6.95 |
| CHUCK DUKE     | 42           | 4.50 |
| TRAVIS EVERS   | 40           | 5.90 |
| TRACY STEELMAN | 45           | 4.00 |
| BETH FORD      | 40           | 4.00 |

The output will contain the Employee's Name, Hours Worked, Rate of Pay, and Gross Pay. If an error is found in the hours worked, write an error message, employee's name and hours worked.

Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## MODULE H (Continued)

### PROGRAM 2

**INSTRUCTIONS:** Modify the Payroll Report to reflect the following changes. Employees who work over 40 hours are paid time-and-a-half as overtime. A deduction of 5% of gross pay is taken out for taxes.

The output will be a five-column report which will contain the Employee's Name, Regular Pay, Overtime Pay, Income Taxes, and Net Pay.

Totals will be taken for Regular Pay, Overtime Pay, Income Taxes, and Net Pay. A count will be taken of the number of employees.

Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## OVERVIEW

### MODULE I: CONTROLLED LOOPS

- I. Loops allow a section of a program to be repeated as many times as needed. A loop contains instructions that will be executed as long as specified conditions are not changed. While in a loop, a condition will be evaluated each time.
  - A. **FOR/NEXT**--a combination of statements used to create a loop.
    1. **FOR** is the keyword beginning the loop.
    2. **NEXT** is the keyword ending the loop. The **NEXT** statement is a key to the computer to increment the index variable; it also checks to see if the final value exceeds the value given in the **FOR** statement.
    3. All statements between **FOR** and **NEXT** are executed each time the loop is repeated.
  - B. The format of the **FOR/NEXT** statement is given in the following example:  
**FOR index variable = initial value TO final value STEP increment value**
    1. **FOR, TO, and STEP**--are the keywords.
    2. **Index variable**--the counter variable which changes each time the loop is repeated.
    3. **Initial value**--the value to start the loop.
    4. **Final value**--the value to end the loop.
    5. **Increment value**--the value by which the index variable is increased.
  - C. All initial, final, or increment values can be either constants or variables. If values are variables, they must be assigned **before** the **FOR** statement is executed.
  - D. If the **STEP** option is omitted, the increment value is assumed to be 1.
  - E. The following are examples in which the **FOR/NEXT** statement can be written:
    1. **FOR J = 1 TO 50 STEP 5**
    2. **FOR H = 1 TO 5**
    3. **FOR S = 1 TO B**

4. FOR L = J TO S STEP 2
  5. FOR N = 16 TO 1 STEP -1
- F. Loops may be written within another loop; these are called "nested loops."
1. Nested loops cannot overlap.
  2. The inner loop must be totally inside the outer loop.

## MODULE 1: CONTROLLED LOOPS

### ACTIVITY 1: PROGRAMMING WITH FOR/NEXT

#### Project 1

**INSTRUCTIONS:** Predict the outcome of the following programs.

1.     10 FOR X = 1 TO 21 STEP 3  
       20 PRINT X,  
       30 NEXT X
2.     20 FOR P = 30 TO 1 STEP -3  
       30 PRINT P;  
       40 NEXT P
3.     40 FOR C = 3 TO 1 STEP -1  
       50 FOR V = 1 TO 4  
       60 PRINT C;" "V  
       70 NEXT V  
       80 NEXT C

#### Project 2

**INSTRUCTIONS:** Debug the following programs.

1.     10 FOR T = 40 TO 20  
       20 PRINT T  
       30 NEXT T
2.     40 FOR B = 1 TO 10  
       50 FOR C = 4 TO 6  
       60 PRINT C,B  
       70 NEXT B  
       80 NEXT C
3.     30 FOR K = 1 TO 15 STEP 3  
       40 IF K = 10 THEN 80  
       50 FOR G = 1 TO 1000 NEXT G  
       60 NEXT K  
       80 PRINT "K = 2"

## MODULE 1

### PROGRAM

**INSTRUCTIONS:** Design, code, key, and execute a program to produce a chart for the current fund-raising project. The chart will list the cost of 1 to 10 products at various prices. The prices will be key to the program through an INPUT statement.

The output will consist of a chart listing the numbers: 1 through 10 in the first column and the amount in the second column. The amount is calculated by multiplying the number of items by the price.

Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## OVERVIEW

### MODULE J: ADVANCED CONCEPTS

- I. **Tables**--are also referred to as arrays. By using a table, a variable may store more than one item of data.
  - A. **Elements**--the term used for data items stored in the table.
  - B. **Subscripts**--tell the row or column that the data item is in the table.
    1. Subscripts are enclosed in parentheses.
    2. A subscript must be a positive whole number.
    3. A subscript may be a constant, a variable, or an arithmetic expression.
  - C. **DIM**--is the keyword used in setting up a table. DIM is short for dimension.
    1. When dimensioning a table that has numeric data, all elements are automatically filled with zeros.
    2. When dimensioning a table that has character data, all elements are automatically filled with spaces.
  - D. **READ** or **INPUT**--are the most common keywords that are used to load the elements into the table.
  - E. **Searching**--is the term used to look up desired information. Most tables are searched sequentially.
- II. **Sorting**--the process of arranging data in numeric or alphabetic order.
  - A. The first element is compared with each element below it.
  - B. The elements are swapped if the first element is larger than the one to which it is being compared.
  - C. This pattern is followed until the end of the table is reached.
  - D. There are several methods of sorting.
- III. Various keywords are used to handle character data.
  - A. **LEFT\$**--is used to handle a given number of characters starting with the first character of the left side.
  - B. **RIGHT\$**--is used to handle a given number of characters starting with the right side.

- C. **MID\$--**is used to handle data at a given point in the string.
  - D. **LEN--**is used to count the number of characters in an item.
  - E. **INKEY\$--**is used to get one character from the keyboard without the user pressing ENTER/RETURN.
  - F. **STR\$--**is used to change numeric data into character data.
  - G. **VAL--**is used to change character data into numeric data.
- IV. **Graphics--**provide a way to produce diagrams or pictures as output. Keywords vary greatly from one computer to the next. Some keywords that are used for an IBM PC would be as follows.
- A. **SCREEN--**selects what mode the display should operate. Computers have different resolution modes.
  - B. **COLOR--**is used to set or change the background and foreground color. Color varies depending on the screen mode selected.
  - C. **PSET--**means point set. PSET activates any specified pixel on the screen.
  - D. **LINE--**is used to draw a line from one point to another point on the screen.
  - E. **CIRCLE--**is used to draw a circle of specified size at any point on the screen. Some computers can draw ellipses and partial circles.
  - F. **DRAW--**is used to make desired shapes by giving the direction of movement and distance.
  - G. **PAINT--**is used to color in a shape.



## MODULE J: ADVANCED CONCEPTS

### ACTIVITIES 1-4: TABLES, SORTING, CHARACTER HANDLING, AND GRAPHICS

**INSTRUCTIONS:** Debug the following table programs.

1.   10 DIM N\$(10)  
      20 N = 10  
      30 FOR X = 1 TO N  
      40 READ N\$(N)  
      50 NEXT X  
      60 FOR A = 1 TO X  
      70 PRINT N\$(X)  
      80 NEXT A  
      1000 DATA KIMBERLY,MICHELLE,BISMARK,ELKE,M.E.,BIFF,  
      CHESTER,JOHN,SUSAN,O.P.
  
2.   40 DIM R\$(4)  
      50 J = 5  
      60 READ N\$(J)  
      70 FOR B = 1 TO J  
      80 PRINT N\$(J)  
      90 NEXT B  
      1000 BUD,CATHY,JIM,MARGARET,SISSY

## MODULE J

### PROGRAM

**INSTRUCTIONS:** Design, code, key, and execute a program to produce a Contestant Listing.

The program will load a table using the INPUT statement. The input will be the contestant name and contest. The information should be sorted alphabetically by contestant name or contest.

The output will consist of a list including the contestant's name and the contest.

Be sure to use proper headings.

Use the proper documentation methods requested by the instructor.

## **SECTION IV**

### **DATA STORAGE**

#### **ESSENTIAL ELEMENT**

**75.87(g)(6)(A)(B)(C)**

6. **Concepts and skills associated with data storage.**

**The student will be provided opportunities to:**

- (A) Demonstrate an understanding of the types of media for data storage;**
- (B) Identify methods of storing data on storage devices; and**
- (C) Create and maintain a program to store and retrieve data from a file.**

## SUGGESTED UNIT LESSON PLAN

### SECTION IV: DATA STORAGE

#### UNIT I: MEDIA AND TYPES OF STORAGE

|                               |                                  |                                    |
|-------------------------------|----------------------------------|------------------------------------|
| <b>ESSENTIAL<br/>ELEMENT:</b> | <b>75.87<br/>(g)(6)(A)(B)(C)</b> | <b>ALLOCATED<br/>TIME: 7 Hours</b> |
|-------------------------------|----------------------------------|------------------------------------|

**UNIT  
OBJECTIVE:** Students will be able to identify primary and secondary storage devices and media.

**STUDENT  
COMPETENCIES:** After completing this unit, the student will be able to:

1. Define primary and secondary storage.
2. Identify secondary storage devices and media.
3. Distinguish between volatile and nonvolatile memory.
4. Differentiate different file organizations and file accesses.

**FOCUS:** To process and store data, one must understand the role of primary and secondary storage. Application requirements will determine the type of secondary storage and method of access.

**DOMAINS:** Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective, domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:** Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. Computer storage (memory)
  - A. First microcomputers
  - B. Today's microcomputers
  - C. Speed of processing
- II. Primary (main) storage
  - A. Memory chips
  - B. BASIC interpreters
  - C. RAM
  - D. ROM
- III. Secondary (auxiliary) storage
  - A. Common types of secondary storage
  - B. Other forms of secondary storage
  - C. On-line and off-line
  - D. Access methods

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

Specific Teaching Activities

1. Secure various types of memory chips for students to examine.
2. Open up a computer so students can view the internal makeup of the computer. If you have two different types of computers in your classroom, open both up for comparison. (NOTE: Do not do this if your computers are sealed.)
3. Secure examples of the following types of media:

Magnetic tape  
Floppy diskette  
Punched card  
Optical disk

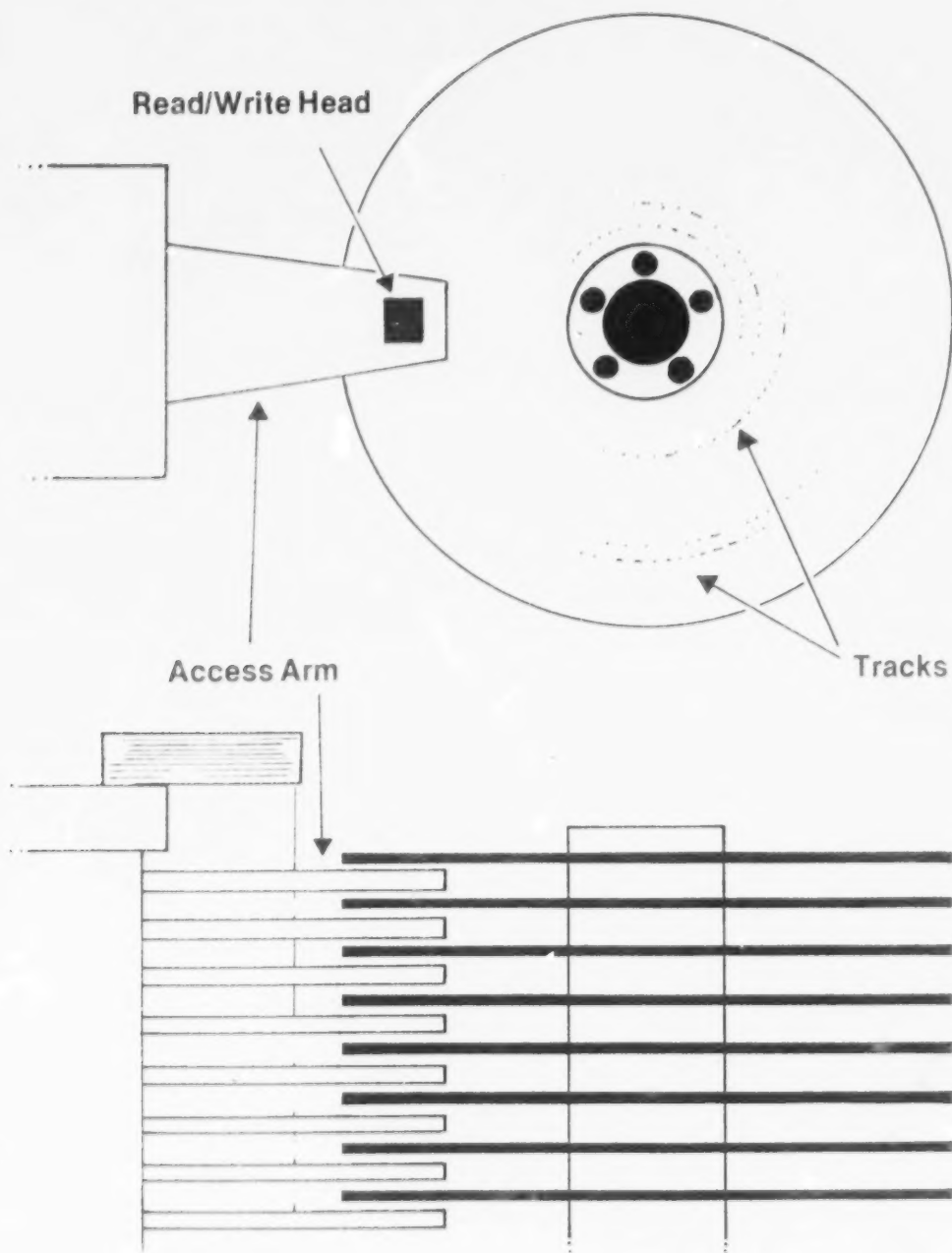
## ACTIVITY MATRIX

[illegible]

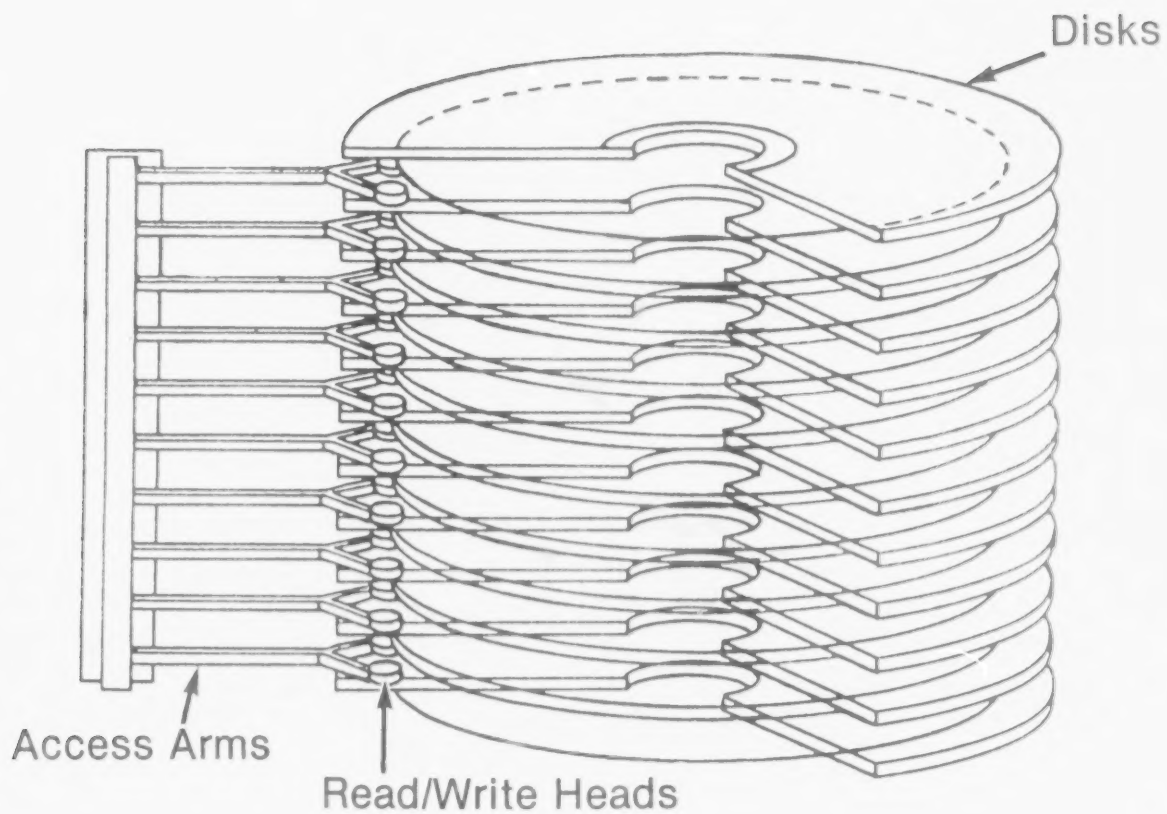


[illegible]

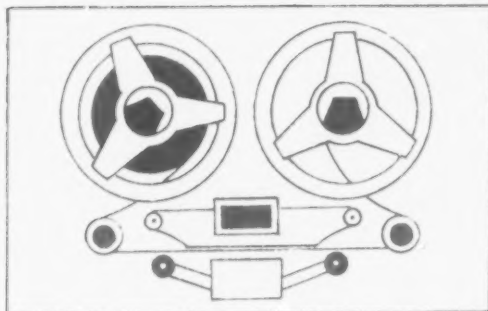
## DISK PACK MECHANISMS



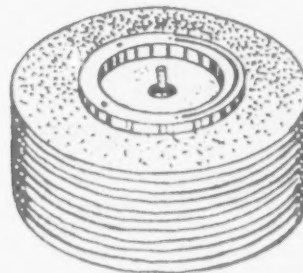
## THE DISK DEVICE



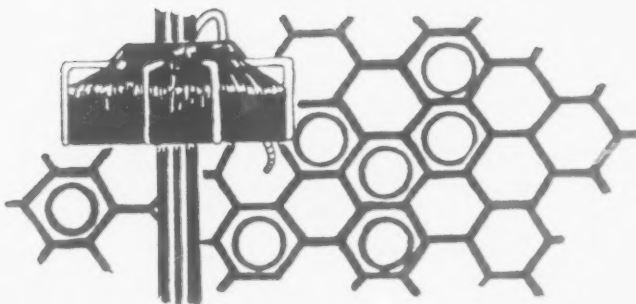
## SECONDARY STORAGE MEDIA



MAGNETIC TAPE

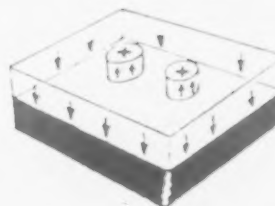


MAGNETIC DISK

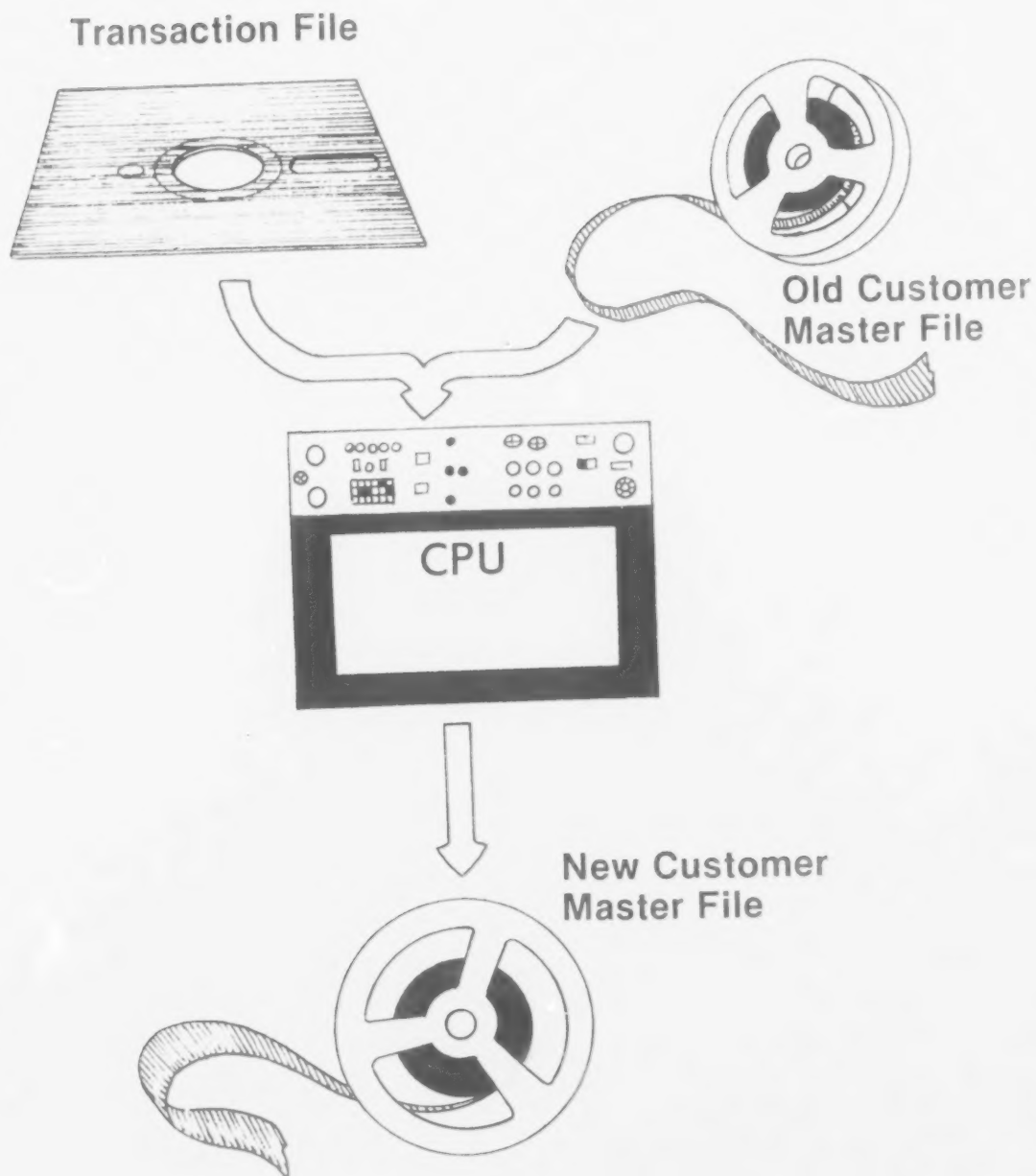


MASS STORAGE

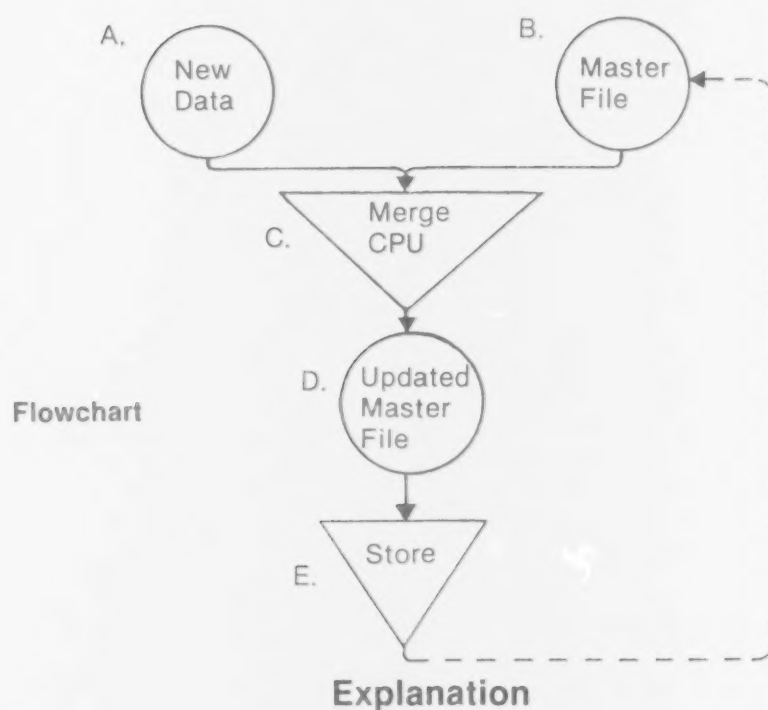
MAGNETIC BUBBLE



## SEQUENTIAL FILE UPDATING

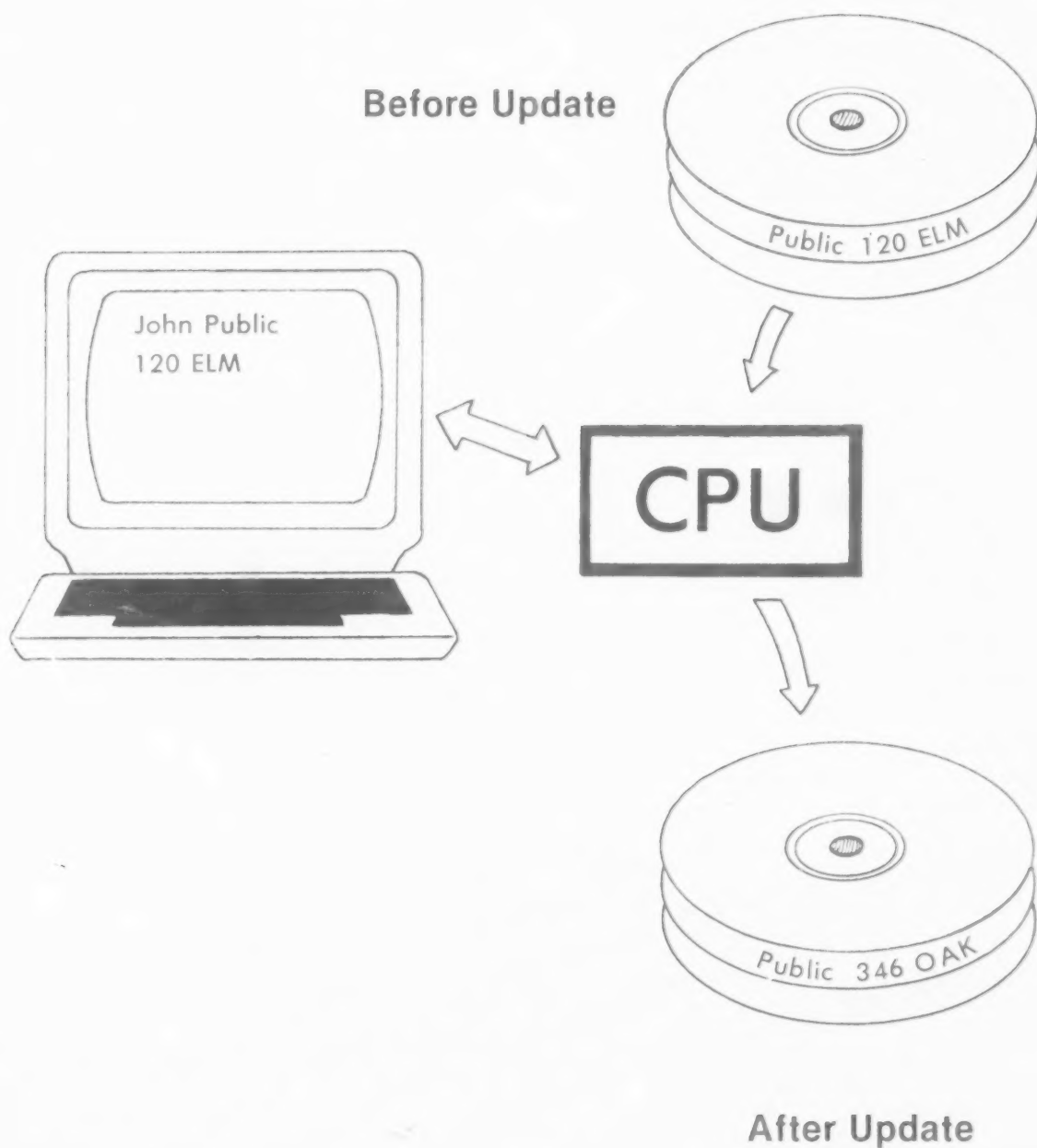


## AN EXAMPLE OF SEQUENTIAL FILE UPDATING



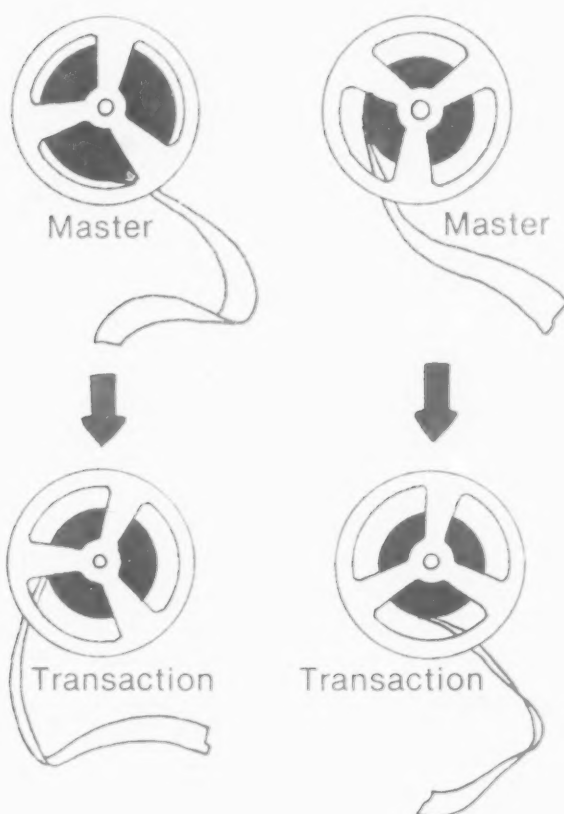
- A. The magnetic tape file of new data and
- B. the master magnetic tape file are
- C. merged (put together in sequence) by the CPU.
- D. A new, updated master file is created and
- E. is stored temporarily. This will be the new file when the data needs to be updated again.

## RANDOM FILE UPDATING



## GRANDFATHER, FATHER, SON BACKUP

### Two Previous



(grandfather)

(father)

### Current Master File



(son)



## COMPETENCY TEST

**INSTRUCTIONS:** Complete each of the following statements by writing in the correct term in the blank space.

1. The memory located inside the computer is called main storage or \_\_\_\_\_ storage.
2. Each position in main memory is referred to as a \_\_\_\_\_.
3. ROM stands for \_\_\_\_\_.
4. RAM stands for \_\_\_\_\_.
5. ROM is said to be \_\_\_\_\_; the information is not erased when the power is turned off.
6. RAM is said to be \_\_\_\_\_ because the information is erased when the power is turned off.
7. Main memory consists of memory \_\_\_\_\_ located inside the CPU.
8. Secondary storage is also referred to as \_\_\_\_\_ storage.
9. The most common forms of secondary storage are magnetic \_\_\_\_\_ and \_\_\_\_\_.
- 10.-13. List four types of secondary storage media: \_\_\_\_\_  
\_\_\_\_\_
14. When secondary storage is communicating with the processor it is said to be \_\_\_\_\_.
15. Information can be stored on secondary media either \_\_\_\_\_ or \_\_\_\_\_.
16. \_\_\_\_\_ access is the method of storage used when processing with magnetic tape.
17. The fastest method of accessing information is \_\_\_\_\_.
18. Batch processing traditionally saves information as a \_\_\_\_\_.
19. Interactive processing needs faster access to information; therefore, it uses the \_\_\_\_\_ method.
20. A disk uses the \_\_\_\_\_ method of access.

## ANSWERS TO ACTIVITIES

### Activity One

1. Main
2. Auxiliary
3. Is also called main storage. It consists of memory chips located in the CPU which vary in the amount of information it can store.
4. Is also called auxiliary storage. Secondary storage is all other forms of storage other than primary.

### Activity Two

- |      |            |
|------|------------|
| 1. B | 5. B       |
| 2. D | 6. B, C, D |
| 3. A | 7. A       |
| 4. C | 8. B       |

### Activity Three

#### Project 1

Volatile evaporates with power off.  
Nonvolatile does not evaporate.

#### Project 2

1. Read only memory
2. Nonvolatile
3. Random-access memory
4. Volatile
5. RAM

### Activity Four

#### Project 1

Random access compares to selecting a song on a record. It goes directly to the song.  
Sequential access compares to selecting a song on a tape.

Project 2 Evaluated to the satisfaction of the instructor.

## COMPETENCY TEST ANSWERS

1. Primary
2. Byte
3. Read-only memory
4. Random-access memory
5. Nonvolatile
6. Volatile
7. Chips
8. Auxiliary
9. Tape and disk
10. Punch cards
11. Magnetic bubble storage
12. Optical disk
13. Magnetic drum (other answers are acceptable)
14. On-line
15. Sequentially or randomly
16. Sequential
17. Random
18. Group
19. On-line
20. Direct

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(6)(A)(B)

Students will be provided opportunities to demonstrate an understanding of the types of media for data storage, identify methods of storing data on storage devices, and create and maintain a program to store and retrieve data from a file.

---

---

**OBJECTIVE:** Students will be able to identify primary and secondary storage devices and media.

|  | MASTERY | NON-MASTERY |
|--|---------|-------------|
|--|---------|-------------|

- |                                                                  |       |       |
|------------------------------------------------------------------|-------|-------|
| 1. Define primary and secondary storage.                         | _____ | _____ |
| 2. Identify secondary storage devices and media.                 | _____ | _____ |
| 3. Distinguish between volatile and nonvolatile memory.          | _____ | _____ |
| 4. Differentiate different file organizations and file accesses. | _____ | _____ |
|                                                                  | _____ | _____ |
|                                                                  | _____ | _____ |

STUDENT NAME \_\_\_\_\_

## SUGGESTED UNIT LESSON PLAN

### SECTION IV: DATA STORAGE

#### UNIT II: DATA STORAGE APPLICATIONS

**ESSENTIAL  
ELEMENT:**

**75.87  
(g)(6)(C)**

**ALLOCATED  
TIME: 8 Hours**

**UNIT  
OBJECTIVE:**

Students will be able to write a program to create and maintain a data file.

**STUDENT  
OBJECTIVES:**

After completing this unit, the student will be able to:

1. Identify data elements--field, record, and file.
2. Code a program to create a sequential and/or random data file.
3. Code a program to retrieve data and produce a report from a data file they created.

**FOCUS:**

When data are to be saved, programs must be developed to store and access the data file.

**DOMAINS:**

Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:**

Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. Files, records, fields
  - A. File
  - B. Record
  - C. Field
  - D. Use of files, records, and fields
- II. Database programs
- III. Planning for a data file
  - A. Which type
  - B. How much data
  - C. How to organize files

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

Specific Teaching Activities

1. Discuss in detail the input/output-related terms, including record, field, and file.
2. In your discussion of this unit, compare and contrast the various database software packages with utilities available on the larger computers.

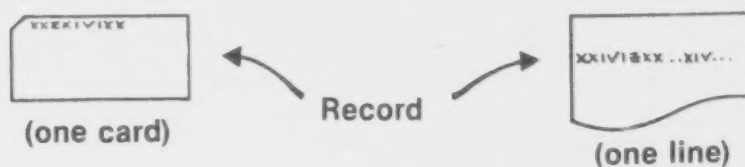




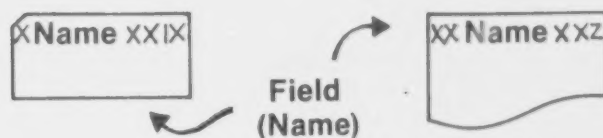
[illegible]

## INPUT/OUTPUT-RELATED TERMS

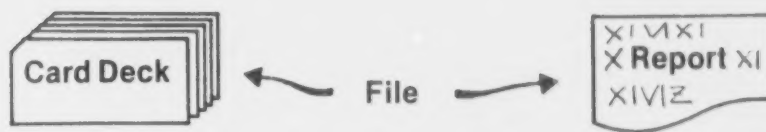
**Record:** One unit of data; e.g., one card or one line on a report.



**Field:** One unit of data within a record; e.g., one item on a card or report.



**File:** A group of related records; e.g., a card deck or a complete report.



BD 101



A Fact

|            |        |
|------------|--------|
| Part Name: | Drill  |
| Part No.:  | BD 101 |
| Price:     | 19.50  |
| Quantity:  | 21     |



A Record

|            |        |
|------------|--------|
| Part Name: | Level  |
| Part Name: | Hammer |
| Part Name: | Drill  |
| Part No.:  | BD 101 |
| Price:     | 19.50  |
| Quantity:  | 21     |



A File

## COMPETENCY TEST

### PART I: DATA ELEMENTS

**INSTRUCTIONS:** Complete each of the following sentences by writing the correct term in the space provided.

1. Define a file:
2. Define a record:
3. Define a field:
4. Give an example showing how a file, a record, and a field are related.
5. Larger computer systems use \_\_\_\_\_ programs to manipulate data files.
6. Much \_\_\_\_\_ must go into setting up a file when determining all the options that are available.
- 7-9. Files can be organized as \_\_\_\_\_, \_\_\_\_\_, or \_\_\_\_\_.
10. When accessing information from a disk, specify the disk \_\_\_\_\_.
- 11-12. The disk can be identified by its \_\_\_\_\_ and \_\_\_\_\_.
13. Records in a relative file are accessed according to a key field and its relative \_\_\_\_\_ in the file.

**BCPI/Data Storage**  
**Unit 2: Data Storage Applications**  
**Teacher's Guide/COMPETENCY TEST**

14. Indexed files locate a record according to an \_\_\_\_\_ based upon the field given.
15. Relative files provide \_\_\_\_\_ access; whereas, indexed files are accessed only through an \_\_\_\_\_.
16. List an advantage of indexed files.
17. List an advantage of relative files.
18. Give an example of a record with three fields. Indicate which field could be designated to be the key field when assigning the relative position in the file.
19. Using the same example as above, which field(s) could be used in an indexed file?

## PART II: SEQUENTIAL DATA FILE

### Project 1

**INSTRUCTIONS:** Write a BASIC program to create a sequential data file containing the fields: invoice number, date, quantity, description, and price. Name the data file *stestxx*, and save it on your data disk. Name the program *seqtxx*. Submit program listing, coding, and data file list to your instructor.

### Project 2

**INSTRUCTIONS:** Write a program to access the data file created (*stestxx*). Headings should include your name, File Display - Test, and main heading. The detail line should include date, invoice number, quantity, description, price, and total (quantity x price). Name the program *aseqxx*. Submit output, program listing, coding, and printer spacing chart to your instructor.

**Input:**

| Invoice # | Date   | Quantity | Description | Price |
|-----------|--------|----------|-------------|-------|
| 3457      | 051290 | 18       | Marigolds   | .25   |
| 3457      | 051290 | 7        | Daisies     | 1.00  |
| 3458      | 051390 | 12       | Thrift      | .50   |
| 3458      | 051390 | 4        | Rose Bush   | 5.99  |
| 3459      | 051590 | 8        | Pecan Tree  | 63.92 |
| 3460      | 051590 | 5        | Petunias    | .88   |

### PART III: DIRECT DATA FILE

#### Project 1

**INSTRUCTIONS:** Write a BASIC program to create a direct data file containing the fields: student ID, name, period 1 grade, period 2 grade, period 3 grade, period 4 grade, period 5 grade, and period 6 grade. Name the data file *dtestxx* on your data disk. Name the program *dirtxx*. Submit your program listing and coding to your instructor

#### Project 2

**INSTRUCTIONS:** Write a program to access the data file created (*dtestxx*). Display student name, ID, each period grade, and each student's final average (grades). Name the program *adirxx*. Your instructor may vary the desired output. Submit output, program listing, coding, and printer spacing chart to your instructor.

**Input:**

| ID     | Name          | 1   | 2  | 3   | 4  | 5   | 6  |
|--------|---------------|-----|----|-----|----|-----|----|
| 788554 | Monica Lerma  | 100 | 85 | 77  | 90 | 88  | 78 |
| 114568 | Maria Flores  | 82  | 74 | 70  | 85 | 91  | 75 |
| 887452 | Michael Moses | 89  | 99 | 73  | 85 | 100 | 74 |
| 123344 | Andre Young   | 67  | 88 | 97  | 70 | 77  | 80 |
| 967332 | Cecilia McKee | 40  | 86 | 85  | 79 | 80  | 95 |
| 564127 | Kathy Thomas  | 90  | 88 | 100 | 76 | 82  | 99 |



## ANSWERS TO ACTIVITIES

### Activity 1

File: A collection of records.  
Record: A unit of information within a file.  
Field: A unit within a record.

### Activity 2

Project 1 Solutions could vary (program written on TRS80/IV). Instructor needs to be familiar with the version of BASIC used on computer.

```
10 REM your name
20 REM sample program to create a sequential file - addxx
30 REM
40 CLS
50 OPEN "O",3,"ADDxx:1"
60 INPUT "Enter Name";N$
70 IF N$ = "END" THEN 150
80 INPUT "ENTER ADDRESS";ADD$
90 INPUT "ENTER CITY";CTY$
100 INPUT "ENTER STATE";ST$
110 INPUT "ENTER ZIP CODE";ZIP:PRINT
120 WRITE #3,N$,ADD$,CTY$,ST$,ZIP
130 REM
140 GOTO 60
150 CLOSE #3
160 END
```

### Project 2

```
10 REM YOUR NAME
20 REM SAMPLE PROGRAM TO ACCESS SEQUENTIAL FILE - ADDXX
30 REM
40 CLS
50 LET F1$="/ / / / / #####"
60 REM HEADINGS
70 PRINT TAB(38) "YOUR NAME":PRINT
80 PRINT TAB(35) "FILE DISPLAY-ADDXX"
90 PRINT:PRINT TAB(37) "ADDRESS LIST"
100 PRINT:PRINT
110 OPEN "I",3,"ADDXX:1"
120 IF EOF(3) THEN 160
130 INPUT #3,N$,ADD$,CTY$,ST$,ZIP
140 PRINT TAB(5) USING F1$; N$,ADD$,CTY$,ST$,ZIP
150 GOTO 120
160 CLOSE #3:END
```

Project 2 (Continued)

Sample output detail line:

Jones Ray 40198 Fallas Rd. Ft. Worth TX 76101

Project 3

```
10 REM YOUR NAME
20 REM SAMPLE PROGRAM TO CREATE A SEQUENTIAL FILE - INVXX
30 REM
40 CLS
50 OPEN "O",2,"INVXX:1"
60 INPUT "ENTER ITEM NUMBER, QUANTITY ON HAND, AND
DESCRIPTION";ITEMNO,OHQTY,DESCR$
70 IF ITEMNO = 99999 THEN 120
80 INPUT "ENTER QUANTITY ON ORDER,U/M,AND
PRICE",OOQTY,UM$,PRICE
90 INPUT "ENTER COST";COST:PRINT
100 WRITE #2, ITEMNO,DESCR$,OHQTY,OOQTY,UM$,PRICE,COST
110 GOTO 60
120 CLOSE #2
130 END
```

Project 4

```
10 REM YOUR NAME
20 REM SAMPLE PROGRAM TO PRINT SEQUENTIAL FILE - INVXX
30 REM
40 LET F1$ = "#,###  ##### /      /  ###.##  $$#,###.##"
50 CLS
60 REM HEADINGS
70 PRINT TAB(38) "YOUR NAME":PRINT
80 PRINT TAB(35) "FILE DISPLAY-INVXX"
90 PRINT:PRINT TAB(37) "INVENTORY LIST"
100 PRINT:PRINT
110 OPEN "I",3,"INVXX:1"
120 IF EOF(3) THEN 170
130 INPUT #3,ITEMNO,DESCR$,OHQTY,OOQTY,UM$,PRICE,COST
140 LET V = OOQTY * COST
150 PRINT TAB(5) USING F1$; OOQTY,ITEMNO,DESCR$,COST,V
160 PRINT:GOTO 120
170 CLOSE #3:END
```

Sample output detail line:

1,500 63219 Dinner Plate 2.52 \$3,780.00

### Project 5

```
10 REM YOUR NAME
20 REM SAMPLE PROGRAM TO CREATE A DIRECT FILE - PAYXX
30 REM
40 CLS
50 OPEN "D",3,"PAYXX:1",39
60 FIELD 3, 10 AS SS$, 20 AS NA$, 2 AS RH$, 3 AS R$, 4 AS D$
70 INPUT "SOCIAL SECURITY NO.,NAME"; S#,N$
80 IF S# = 99 THEN 170
90 INPUT "REGULAR HOURS, RATE OF PAY"; RH, RATE
100 INPUT "DEDUCTIONS"; DED
110 INPUT "RECORD NUMBER"; REC%
120 PRINT
130 LSET SS$=STR$(S#): LSET NA$=N$:LSET RH$=MKI$(RH)
140 LSET R$=MKI$(RATE):LSET D$=MKI$(DED)
150 PUT 3, REC%
160 GOTO 70
170 CLOSE 3
180 END
```

### Project 6

```
10 REM YOUR NAME
20 REM SAMPLE PROGRAM TO ACCESS DIRECT FILE - PAYXX
30 REM
40 OPEN "D",1,"PAYXX:1",39
50 FIELD 1,10 AS SS$, 20 AS NA$, 2 AS RH$, 3 AS R$, 4 AS D$
60 CLS
70 INPUT "RECORD NO. DESIRED"; REC%
80 IF REC% = 0 THEN 150
85 GET 1, REC%
90 PRINT USING "#####"; VAL(SS$)
100 PRINT NA$
110 PRINT USING "##"; CVI(RH$)
120 PRINT USING "$$##.##"; CVI(RH$) * CVI(R$) - CVI(D$)
130 PRINT
140 GOTO 70
150 CLOSE 1: END
```

Sample output detail line:

```
RECORD NO. DESIRED? 3
005099428
JOHN SWAN
15
$56.00
```

### Activity 3

#### Project 1

```
10 REM YOUR NAME
20 REM SAMPLE PROGRAM TO CREATE A DIRECT FILE - SALEXX
30 REM
40 CLS
50 OPEN "D",2,"SALEXX:1",32
60 FIELD 2, 4 AS QTY$, 20 AS DESC$, 5 AS PROD$, 3 AS UNIT$
70 INPUT "QUANTITY, DESCRIPTION",QTY,DESCRIP$
80 IF QTY = 0 THEN 160
90 INPUT "PRODUCT NO.,UNIT PRICE"; PROD,PRICE
100 INPUT "RECORD NUMBER"; REC%
110 LSET QTY$ = MKI$(QTY); LSET DESC$ = DESCRIP$
120 LSET PROD$ = MKS$(PROD); LSET UNIT$ = MKI$(PRICE)
130 PUT 2, REC%
140 PRINT
150 GOTO 70
160 CLOSE 2
170 END
```

#### Project 2

```
10 REM YOUR NAME
20 REM SAMPLE PROGRAM TO ACCESS DIRECT FILE - SALEXX
30 REM
40 CLS
50 OPEN "D",1,"SALEXX:1",32
60 FIELD 1,4 AS QTY$, 20 AS DESC$, 5 AS PROD$, 3 AS UNIT$
70 INPUT "RECORD NO. REQUESTED"; REC%
80 IF REC% = 0 THEN 160
90 GET 1, REC%
100 PRINT USING "####"; CVI(QTY$)
110 PRINT USING "#.###"; (CVI(UNIT$))/100
120 PRINT DESC$
130 PRINT USING "$$#.##"; (CVI(QTY$) * CVI(UNIT$))/100
140 PRINT
150 GOTO 70
160 CLOSE 1:END
```

Sample output detail line:

RECORD NUMBER? 1

61  
7.30  
SALT  
\$445.30

## COMPETENCY TEST ANSWERS

### Part I

1. A collection of records.
2. A unit of information within a file.
3. A unit within a record.
4. File--a collection of student records.  
Record--one student record.  
Field--address or name field.
5. Utility
6. Planning
7. Random
8. Sequential
9. Indexed
10. Address
11. Track
12. Sector number
13. Position
14. Index
15. Direct, index
16. Able to access files by different fields.
17. Fast access.
18. Evaluated to the satisfaction of the instructor.
19. Evaluated to the satisfaction of the instructor.

### Part II

#### Project 1

```
10 REM YOUR NAME
20 REM TEST PROGRAM TO CREATE A SEQUENTIAL FILE - STSTXX
30 REM
40 CLS
50 CLS
60 OPEN "O",2,"STSTXX:1"
70 INPUT "ENTER INVOICE NO."; NO
80 IF NO = 0 THEN 160
90 INPUT "ENTER DATE"; DATE
100 INPUT "ENTER QUANTITY"; QUAN
110 INPUT "ENTER DESCRIPTION"; DES$
120 INPUT "ENTER PRICE"; PRICE
130 PRINT
140 WRITE #2, NO,DATE,QUAN,DES$,PRICE
150 GOTO 70
160 CLOSE #2
170 END
```

## Project 2

```
10 REM YOUR NAME
20 REM TEST PROGRAM TO ACCESS SEQUENTIAL FILE - STSTXX
30 REM
40 CLS
50 LET F1$ = "#####  ####  ## /      / ##.## $$#.##"
60 REM HEADINGS
70 PRINT TAB(38) "YOUR NAME":PRINT
80 PRINT TAB(35) "FILE DISPLAY - STSTXX"
90 PRINT:PRINT TAB(37) "INVOICE LIST"
100 PRINT:PRINT
110 OPEN "I",2,"STSTXX:1"
120 IF EOF(2) THEN 170
130 INPUT #2, NO,DATE,QUAN,DESS,PRICE
140 LET TOT = QUAN * PRICE
150 PRINT TAB(21) USING F1$: DATE,NO,QUAN,DESS,PRICE,TOT
160 GOTO 120
170 CLOSE #2
180 END
```

Sample output detail line:

051290 3457 18 Marigolds 0.25 \$4.50

## Part III

### Project 1

```
10 REM YOUR NAME
20 REM TEST PROGRAM TO CREATE DIRECT FILE = DTESTXX
30 REM
40 CLS
50 OPEN "D",3,"DTESTXX:1",44
60 FIELD 3,6 AS STUDENT$, 20 AS NA$, 3 AS P1$, 3 AS P2$, 3 AS P3$, 3 AS
P4$, 3 AS P5$, 3 AS P6$
70 INPUT "STUDENT ID, AND NAME"; ID, NAM$
80 IF ID = 0 THEN 220
90 INPUT "PERIOD 1 GRADE";P1
100 INPUT "PERIOD 2 GRADE";P2
110 INPUT "PERIOD 3 GRADE";P3
120 INPUT "PERIOD 4 GRADE";P4
130 INPUT "PERIOD 5 GRADE";P5
140 INPUT "PERIOD 6 GRADE";P6
150 INPUT "RECORD NUMBER";REC%
160 LSET STUDENT$ = MK$(ID): LSET NA$=NAM$: LSET P1$=MK$(P1)
170 LSET P2$=MK$(P2): LSET P3$=MK$(P3): LSET P4$=MK$(P4)
180 LSET P5$=MK$(P5): LSET P6$=MK$(P6)
190 PUT 3, REC%
```

BCPI/Data Storage  
Unit 2: Data Storage Applications  
Teacher's Guide/ANSWERS

```
200 PRINT
210 GOTO 70
220 CLOSE 3
240 END
```

Project 2

```
10 REM YOUR NAME
20 REM TEST PROGRAM TO ACCESS DIRECT FILE - DTESTXX
30 REM
40 CLS
50 OPEN "D",2,"DTESTXX:1",44
60 FIELD 2, 6 AS STUDENT$, 20 AS NA$, 3 AS P1$, 3 AS P2$, 3 AS P3$, 3 AS
P4$, 3 AS P5$, 3 AS P6$
70 INPUT "REQUEST RECORD NO."; REC%
80 IF REC% = 0 THEN 160
90 GET 2, REC%
100 PRINT:PRINT NA$
110 PRINT USING "#####"; CVS(STUDENT$)
120 PRINT "FINAL AVERAGE",
130 PRINT USING "###.##"; (CVI(P1$) + CVI(P2$) + CVI(P3$) + CVI(P4$) +
CVI(P5$) + CVI(P6$))/6
140 PRINT
150 GOTO 70
160 CLOSE 2
170 END
```

Sample output detail line:

REQUEST RECORD NO.? 2

Maria Flores

114568

FINAL AVERAGE 79.50

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(6)(C)

Students will be able to create and maintain a program to store and retrieve data from a file.

| OBJECTIVE: | Students will be able to write<br>a program to create and maintain<br>a data file. | MASTERY | NON-<br>MASTERY |
|------------|------------------------------------------------------------------------------------|---------|-----------------|
|------------|------------------------------------------------------------------------------------|---------|-----------------|

- |                                                                                        |       |       |
|----------------------------------------------------------------------------------------|-------|-------|
| 1. Identify data elements--field, record, and file.                                    | _____ | _____ |
| 2. Code a program to create a sequential and/or random data file.                      | _____ | _____ |
| 3. Code a program to retrieve data and produce a report from a data file they created. | _____ | _____ |
|                                                                                        | _____ | _____ |
|                                                                                        | _____ | _____ |
|                                                                                        | _____ | _____ |

592

STUDENT NAME \_\_\_\_\_



## SUGGESTED UNIT LESSON PLAN

### SECTION IV: DATA STORAGE

#### UNIT III: DATA REPRESENTATION AND NUMBERING SYSTEMS

**ESSENTIAL  
ELEMENT:** 75.87  
(g)(6)

**ALLOCATED  
TIME:** 5 Hours

**UNIT  
OBJECTIVE:** Students will be able to demonstrate knowledge of computer numbering systems.

**STUDENT  
COMPETENCIES:** After completing this unit, the student will be able to:

1. Define character and control codes.
2. Give examples of ASCII coding.
3. Give examples of EBCDIC coding.
4. Demonstrate an understanding of the binary numbering system.
5. Demonstrate an understanding of the octal numbering system.
6. Demonstrate an understanding of the hexadecimal numbering system.

**FOCUS:** In order to create and manipulate data files, one should understand data representation as it relates to primary and secondary storage.

**DOMAINS:** Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:** Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. ASCII code
  - A. Numbers ASCII uses
  - B. Nonprinting, or control, characters
  - C. Screen characters
  - D. Widely used
- II. EBCDIC code
  - A. Numbers EBCDIC uses
  - B. ASCII and EBCDIC communicate
- III. Binary code
  - A. "On" (1) and "off" (0)
  - B. Define bit
  - C. Nibble
  - D. ASCII and EBCDIC converts to binary
  - E. Decimal numbering system
  - F. Binary numbering system
  - G. Octal numbering system
  - H. Hexadecimal numbering system
  - I. Reading a bit
- IV. Parity bit
  - A. Self-checking
  - B. Odd or even number of bits
  - C. Even parity
  - D. Understanding computer operations
  - E. Understanding code

## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview the unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

Specific Teaching Activities

1. Please note that this unit does not have a specific essential element assigned; however, this unit is useful and should be incorporated in this section.
2. Prepare charts showing formulas used for conversion of decimal, binary, octal, and hexadecimal numbers.
3. Work through each of the numbering systems with the students. You may wish to provide several other numbering problems for students to have additional practice in conversions.

[illegible]

## COMPETENCY TEST MATRIX

[illegible]

### EBCDIC AND ASCII CODES

| Bit Positions in EBCDIC |   |   |   |              |   |   |   | Character |
|-------------------------|---|---|---|--------------|---|---|---|-----------|
| Zone Bits               |   |   |   | Numeric Bits |   |   |   |           |
| 1                       | 1 | 1 | 1 | 0            | 1 | 0 | 1 | 5         |
| 1                       | 1 | 0 | 0 | 0            | 0 | 0 | 1 | A         |
| 0                       | 1 | 0 | 1 | 1            | 0 | 1 | 1 | \$        |

EBCDIC Code for Characters 5, A, \$

| Bit Positions in ASCII |   |   |         |   |   |   | Character |
|------------------------|---|---|---------|---|---|---|-----------|
| Zone                   |   |   | Numeric |   |   |   |           |
| 0                      | 1 | 1 | 0       | 1 | 0 | 1 | 5         |
| 1                      | 0 | 0 | 0       | 0 | 0 | 1 | A         |

ASCII Code for Characters 5 and A

**EXTENDED BINARY CODED DECIMAL INTERCHANGE CODE  
(EBCDIC)**

---

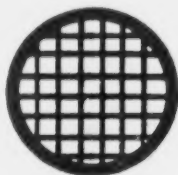
| <i>Character</i> | <i>EBCDIC</i> | <i>Character</i> | <i>EBCDIC</i> |
|------------------|---------------|------------------|---------------|
| 0.....           | 1111 0000     | I.....           | 1100 1001     |
| 1.....           | 1111 0001     | J.....           | 1101 0001     |
| 2.....           | 1111 0010     | K.....           | 1101 0010     |
| 3.....           | 1111 0011     | L.....           | 1101 0011     |
| 4.....           | 1111 0100     | M.....           | 1101 0100     |
| 5.....           | 1111 0101     | N.....           | 1101 0101     |
| 6.....           | 1111 0110     | O.....           | 1101 0110     |
| 7.....           | 1111 0111     | P.....           | 1101 0111     |
| 8.....           | 1111 1000     | Q.....           | 1101 1000     |
| 9.....           | 1111 1001     | R.....           | 1101 1001     |
| A.....           | 1100 0001     | S.....           | 1110 0010     |
| B.....           | 1100 0010     | T.....           | 1110 0011     |
| C.....           | 1100 0011     | U.....           | 1110 0100     |
| D.....           | 1100 0100     | V.....           | 1110 0101     |
| E.....           | 1100 0101     | W.....           | 1110 0110     |
| F.....           | 1100 0110     | X.....           | 1110 0111     |
| G.....           | 1100 0111     | Y.....           | 1110 1000     |
| H.....           | 1100 1000     | Z.....           | 1110 1001     |

---



## BINARY NUMBERING SYSTEM (BASE 2)

Place Value: 8



on

1

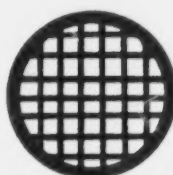
4



off

0

2



on

1

1



off

0

Binary Number 1010 = Decimal Number 10

(8 + 2)

### BINARY TABLES

|            |                     |                    |                   |                   |                     |
|------------|---------------------|--------------------|-------------------|-------------------|---------------------|
| Number 1 = | 0<br>Eight<br>$2^3$ | 0<br>Four<br>$2^2$ | 0<br>Two<br>$2^1$ | 1<br>One<br>$2^0$ | Place Value<br>Base |
| Number 2 = | 0<br>Eight<br>$2^3$ | 0<br>Four<br>$2^2$ | 1<br>Two<br>$2^1$ | 0<br>One<br>$2^0$ | Place Value<br>Base |

Table 1

| Decimal Digit | Place Value |   |   |   | Binary<br>Equivalent |
|---------------|-------------|---|---|---|----------------------|
|               | 8           | 4 | 2 | 1 |                      |
| 0             | 0           | 0 | 0 | 0 | 0000                 |
| 1             | 0           | 0 | 0 | 1 | 0001                 |
| 2             | 0           | 0 | 1 | 0 | 0010                 |
| 3             | 0           | 0 | 1 | 1 | 0011                 |
| 4             | 0           | 1 | 0 | 0 | 0100                 |
| 5             | 0           | 1 | 0 | 1 | 0101                 |
| 6             | 0           | 1 | 1 | 0 | 0110                 |
| 7             | 0           | 1 | 1 | 1 | 0111                 |
| 8             | 1           | 0 | 0 | 0 | 1000                 |
| 9             | 1           | 0 | 0 | 1 | 1001                 |

Table 2

## CHARACTERISTICS OF NUMBERING SYSTEMS

### Decimal

**Base 10**      10 digit choices  
0, 1, 2, 3, 4, 5, 6, 7, 8, 9  
largest single digit is 9

### Binary

**Base 2**      2 digit choices  
0, 1  
largest single digit is 1

### Octal

**Base 8**      8 digit choices  
0, 1, 2, 3, 4, 5, 6, 7  
largest single digit is 7

### Hexadecimal

**Base 16**      16 digit choices  
0, 1, 2, 3, 4, 5, 6, 7, 8, 9,  
A, B, C, D, E, F  
largest single digit is F (15)

Digit choices always equal base.

Largest single digit always one less than base.

TM-5

# The Decimal System

## (Base 10)

THE 10 ALLOWABLE CHARACTERS ARE :  
0,1,2,3,4,5,6,7,8,9

| Thousands | Hundreds | Tens   | Ones   | Places          |
|-----------|----------|--------|--------|-----------------|
| $10^3$    | $10^2$   | $10^1$ | $10^0$ | POWERS          |
| 1000      | 100      | 10     | 1      | VALUES          |
|           |          | 1      | 3      | number<br>is 13 |

$$1(10) + 3(1) = 13$$

### 4-BIT AND 6-BIT BCD CODES

| 4    | 3    | 7    | Decimal   |
|------|------|------|-----------|
| 0100 | 0011 | 0111 | 4-Bit BCD |

4-Bit BCD Code for Decimal Number 437

| BCD Position |   |   |   |   |   |   | Character |
|--------------|---|---|---|---|---|---|-----------|
| C            | B | A | 8 | 4 | 2 | 1 |           |
| 1            | 0 | 0 | 0 | 0 | 1 | 1 | 3         |
| 0            | 1 | 1 | 0 | 0 | 0 | 1 | A         |

6-Bit BCD Code for Decimal  
 3 and Character A

# OCTAL EQUIVALENTS FOR DECIMAL NUMBERS

| Decimal | Octal         |              |              |
|---------|---------------|--------------|--------------|
|         | $8^2$<br>(64) | $8^1$<br>(8) | $8^0$<br>(1) |
| 0       |               |              | 0            |
| 1       |               |              | 1            |
| 2       |               |              | 2            |
| 3       |               |              | 3            |
| 4       |               |              | 4            |
| 5       |               |              | 5            |
| 6       |               |              | 6            |
| 7       |               |              | 7            |
| 8       |               | 1            | 0            |
| 9       |               | 1            | 1            |
| 10      |               | 1            | 2            |
| 11      |               | 1            | 3            |
| 12      |               | 1            | 4            |
| 13      |               | 1            | 5            |
| 14      |               | 1            | 6            |
| 15      |               | 1            | 7            |
| 20      |               | 2            | 4            |
| 64      | 1             | 0            | 0            |

← 8 + 3 = 11

← 8 + 7 = 15

← 16 + 4 = 20

← 64 + 0 + 0 = 64

### THE HEXADECIMAL NUMBERING SYSTEM (BASE 16)

| HEXADECIMAL | BINARY | DECIMAL |
|-------------|--------|---------|
| 0           | 0000   | 0       |
| 1           | 0001   | 1       |
| 2           | 0010   | 2       |
| 3           | 0011   | 3       |
| 4           | 0100   | 4       |
| 5           | 0101   | 5       |
| 6           | 0110   | 6       |
| 7           | 0111   | 7       |
| 8           | 1000   | 8       |
| 9           | 1001   | 9       |
| A           | 1010   | 10      |
| B           | 1011   | 11      |
| C           | 1100   | 12      |
| D           | 1101   | 13      |
| E           | 1110   | 14      |
| F           | 1111   | 15      |

## COMPARISON OF NUMBERING SYSTEM PLACE VALUES

### General

| BASE <sup>4</sup> | BASE <sup>3</sup> | BASE <sup>2</sup> | BASE <sup>1</sup> | BASE <sup>0</sup> |
|-------------------|-------------------|-------------------|-------------------|-------------------|
|-------------------|-------------------|-------------------|-------------------|-------------------|

### Base 10

|        |        |        |        |        |
|--------|--------|--------|--------|--------|
| $10^4$ | $10^3$ | $10^2$ | $10^1$ | $10^0$ |
| 10,000 | 1,000  | 100    | 10     | 1      |

### Base 2

|       |       |       |       |       |
|-------|-------|-------|-------|-------|
| $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| 16    | 8     | 4     | 2     | 1     |

### Base 8

|       |       |       |       |       |
|-------|-------|-------|-------|-------|
| $8^4$ | $8^3$ | $8^2$ | $8^1$ | $8^0$ |
| 4,096 | 512   | 64    | 8     | 1     |

### Base 16

|        |        |        |        |        |
|--------|--------|--------|--------|--------|
| $16^4$ | $16^3$ | $16^2$ | $16^1$ | $16^0$ |
| 65,536 | 4,096  | 256    | 16     | 1      |



### CONVERSION TO DECIMAL (1)

**Binary Number:** 1100 = ?

**Step 1.** Determine column values

|   |   |   |   |
|---|---|---|---|
| 8 | 4 | 2 | 1 |
| 1 | 1 | 0 | 0 |

**Step 2.** Multiply column values times column digits =

|           |           |           |           |
|-----------|-----------|-----------|-----------|
| 8         | 4         | 2         | 1         |
| <u>x1</u> | <u>x1</u> | <u>x0</u> | <u>x0</u> |
| 8         | 4         | 0         | 0         |

**Step 3.** Add the products     $8 + 4 + 0 + 0 = 12$

---

**Octal Number:** 146 = ?

**Step 1.** Determine column values

|    |   |   |
|----|---|---|
| 64 | 8 | 1 |
| 1  | 4 | 6 |

**Step 2.** Multiply column values times column digits =

|           |           |           |
|-----------|-----------|-----------|
| 64        | 8         | 1         |
| <u>x1</u> | <u>x4</u> | <u>x6</u> |
| 64        | 32        | 6         |

**Step 3.** Add the products     $64 + 32 + 6 = 102$

## CONVERSION TO DECIMAL (2)

# Conversion to Decimal<sup>2</sup>

**Hexadecimal Number: 2A4D = ?**

**Step 1.** Determine column values

|       |     |    |    |
|-------|-----|----|----|
| 4,096 | 256 | 16 | 1  |
| 2     | 10  | 4  | 13 |

**Step 2.** Multiply column values times column digits

|           |            |           |            |
|-----------|------------|-----------|------------|
| 4,096     | 256        | 16        | 1          |
| <u>x2</u> | <u>x10</u> | <u>x4</u> | <u>x13</u> |
| 8,192     | 2,560      | 64        | 13         |

**Step 3.** Add the products

$$8,192 + 2,560 + 64 + 13 = 10,829$$

## COMPETENCY TEST

### PART I: CONTROL CODES

**INSTRUCTIONS:** Define a character code and a control code; then complete the following sentences.

1. Character code:
  
  
  
  
  
  
  
  
  
  
2. Control code:
  
  
  
  
  
  
  
  
  
  
3. The two most common internal codes are \_\_\_\_\_ and \_\_\_\_\_.
  
  
  
  
  
  
  
  
  
  
4. Microcomputers use the \_\_\_\_\_ code.
  
  
  
  
  
  
  
  
  
  
5. Larger computer systems use the \_\_\_\_\_ code.

## PARTS II & III: ASCII AND EBCDIC CODING

INSTRUCTIONS: Explain major characteristics of ASCII and EBCDIC code.

- 1.
- 2.
- 3.

## PART IV: BINARY NUMBER SYSTEM

INSTRUCTIONS: Convert the following binary numbers to decimal numbers and complete the statements that follow.

1. 0111001 \_\_\_\_\_
2. 11100 \_\_\_\_\_
3. 1111 \_\_\_\_\_
4. 10101 \_\_\_\_\_
5. 1011 \_\_\_\_\_
6. The decimal numbering system uses the base \_\_\_\_\_.
7. The binary numbering system uses the base \_\_\_\_\_.

## PART V: OCTAL NUMBER SYSTEM

**INSTRUCTIONS:** Convert the following octal numbers to decimal numbers, then fill in the answer to the question.

1. 0024 \_\_\_\_\_
2. 00750 \_\_\_\_\_
3. 00400 \_\_\_\_\_
4. 2000 \_\_\_\_\_
5. The octal numbering system uses the base \_\_\_\_\_.

## PART VI: HEXADECIMAL NUMBERS

**INSTRUCTIONS:** Convert the following Hexadecimal numbers to decimal numbers; then complete the statements below.

1. E \_\_\_\_\_
2. 001A \_\_\_\_\_
3. 011 \_\_\_\_\_
4. 111 \_\_\_\_\_
5. 0AB \_\_\_\_\_
6. The hexadecimal numbering system uses the base \_\_\_\_\_.
7. The hexadecimal numbering system uses numbers as well as \_\_\_\_\_.

## PART VII: SUMMARY PROBLEM AND QUESTIONS FOR SECTION IV

### Project 1

**INSTRUCTIONS:** Convert the following decimal numbers to binary, octal, and hexadecimal numbers.

|       | Binary | Octal | Hexadecimal |
|-------|--------|-------|-------------|
| 1. 16 |        |       |             |
| 2. 33 |        |       |             |
| 3. 8  |        |       |             |
| 4. 11 |        |       |             |
| 5. 64 |        |       |             |

## PART VII (Continued)

### Project 2

**INSTRUCTIONS:** Indicate the correct answer by selecting the most appropriate answer and placing the letter in the blank.

- \_\_\_\_\_ 1. The most popular media for auxiliary storage is:
- |         |                         |
|---------|-------------------------|
| A. CPU  | C. ROM                  |
| B. Disk | D. Mass storage devices |
- \_\_\_\_\_ 2. Another term for auxiliary storage is:
- |            |              |
|------------|--------------|
| A. Primary | C. Main      |
| B. Serial  | D. Secondary |
- \_\_\_\_\_ 3. Primary storage is located in the:
- |              |                            |
|--------------|----------------------------|
| A. Processor | C. Central processing unit |
| B. CRT       | D. None of the above       |
- \_\_\_\_\_ 4. Primary storage is measured by the amount of:
- |          |                     |
|----------|---------------------|
| A. Cells | C. Addresses        |
| B. K     | D. All items listed |
- \_\_\_\_\_ 5. When auxiliary storage devices are connected to the computer, they are said to be:
- |             |                 |
|-------------|-----------------|
| A. Off-line | C. On-line      |
| B. Down     | D. Transmitting |
- \_\_\_\_\_ 6. Magnetic tape uses this method for storing and accessing data:
- |               |                 |
|---------------|-----------------|
| A. Sequential | C. Random       |
| B. Direct     | D. Both B and C |
- \_\_\_\_\_ 7. A file consists of numerous:
- |           |                     |
|-----------|---------------------|
| A. Fields | C. Records          |
| B. Disks  | D. Source documents |
- \_\_\_\_\_ 8. A record is further subdivided into:
- |           |           |
|-----------|-----------|
| A. Fields | C. Files  |
| B. Items  | D. Groups |



BCPI/Data Storage  
Unit 3: Data Representation and Numbering Systems  
Teacher's Guide/COMPETENCY TEST

- \_\_\_\_ 9. Files can be organized as:
- |               |                     |
|---------------|---------------------|
| A. Sequential | C. Indexed          |
| B. Relative   | D. All of the above |
- \_\_\_\_ 10. Relative files are accessed according to:
- |             |                  |
|-------------|------------------|
| A. Position | C. Key field     |
| B. Index    | D. Record number |
- \_\_\_\_ 11. Data stored on magnetic disks are recorded in concentric circles called:
- |             |            |
|-------------|------------|
| A. Channels | C. Sectors |
| B. Tracks   | D. Reels   |
- \_\_\_\_ 12. Most small computers use which code for data transmission:
- |           |                |
|-----------|----------------|
| A. EBCDIC | C. Binary      |
| B. ASCII  | D. Hexadecimal |
- \_\_\_\_ 13. The base number for the octal numbering system is:
- |       |      |
|-------|------|
| A. 16 | C. 8 |
| B. 10 | D. 2 |
- \_\_\_\_ 14. A byte traditionally consists of:
- |           |            |
|-----------|------------|
| A. 4 bits | C. 16 bits |
| B. 8 bits | D. 10 bits |
- \_\_\_\_ 15. A parity error involves a problem in data transmission.
- |         |          |
|---------|----------|
| A. True | B. False |
|---------|----------|
- \_\_\_\_ 16. The hexadecimal numbering system utilizes the digits 0-9 and the letters A-F.
- |         |          |
|---------|----------|
| A. True | B. False |
|---------|----------|
- \_\_\_\_ 17. It is impossible for an ASCII computer to communicate with an EBCDIC computer.
- |         |          |
|---------|----------|
| A. True | B. False |
|---------|----------|
- \_\_\_\_ 18. An indexed file has a slower access speed, because it must refer to an appropriate index when searching a file.
- |         |          |
|---------|----------|
| A. True | B. False |
|---------|----------|

**BCPI/Data Storage**  
**Unit 3: Data Representation and Numbering Systems**  
**Teacher's Guide/COMPETENCY TEST**

- \_\_\_\_\_ 19. Relative files have a slower access speed than sequential files.  
A. True B. False
- \_\_\_\_\_ 20. Manipulation of large data files is best handled by database systems.  
A. True B. False
- \_\_\_\_\_ 21. RAM memory is nonvolatile.  
A. True B. False
- \_\_\_\_\_ 22. Volatile means that the information is saved even when the power is turned off.  
A. True B. False
- \_\_\_\_\_ 23. Magnetic disks use the random or direct access to retrieve information.  
A. True B. False
- \_\_\_\_\_ 24. More memory can be added by using secondary storage devices.  
A. True B. False
- \_\_\_\_\_ 25. Data on a disk can be addressed by its track and sector number.  
A. True B. False

## ANSWERS TO ACTIVITIES

### Activity 1

- |          |          |           |
|----------|----------|-----------|
| 1. True  | 6. False | 11. True  |
| 2. False | 7. True  | 12. False |
| 3. False | 8. True  | 13. True  |
| 4. True  | 9. False | 14. True  |
| 5. True  | 10. True | 15. True  |

### Activity 2

#### Project 1

- |              |                |
|--------------|----------------|
| 1. ASCII     | 6. BCD         |
| 2. Bit       | 7. EBCDIC      |
| 3. Binary    | 8. Byte        |
| 4. Parity    | 9. Hexadecimal |
| 5. Check bit | 10. Octal      |

#### Project 2

- |      |       |
|------|-------|
| 1. 2 | 6. 8  |
| 2. 9 | 7. 10 |
| 3. 3 | 8. 1  |
| 4. 5 | 9. 7  |
| 5. 4 | 10. 6 |

### Activity 3

#### Project 1

- |        |
|--------|
| 1. 13  |
| 2. 145 |
| 3. 6   |
| 4. 100 |

#### Project 2

- |             |
|-------------|
| 1. 00011001 |
| 2. 00110010 |
| 3. 1100     |
| 4. 10000010 |

BCPI/Data Storage  
Unit 3: Data Representation and Numbering Systems  
Teacher's Guide/ANSWERS

**Activity 4**

**Project 1**

- |    |    |
|----|----|
| 1. | 10 |
| 2. | 61 |
| 3. | 23 |
| 4. | 12 |

**Project 2**

- |    |      |
|----|------|
| 1. | 006  |
| 2. | 0014 |
| 3. | 0302 |
| 4. | 0604 |

**Activity 5**

**Project 1**

- |    |    |
|----|----|
| 1. | 15 |
| 2. | 5  |
| 3. | 35 |
| 4. | 50 |

**Project 2**

- |    |     |
|----|-----|
| 1. | 78  |
| 2. | C   |
| 3. | 100 |
| 4. | 7   |

620

## COMPETENCY TEST ANSWERS

### Part I

1. Character code--can be seen on the computer screen.
2. Control code--cannot be seen on the screen, because they are concerned with the operation of the machine.
3. ASCII, EBCDIC
4. ASCII
5. EBCDIC

### Parts II & III

1. Use characters 0-128 or 0-256.
2. Used for data communications.
3. Can communicate between each other.

### Part IV

1. 57
2. 29
3. 15
4. 23
5. 11
6. 10
7. 2

### Part V

1. 20
2. 488
3. 256
4. 5632
5. 8

### Part VI

1. 14
2. 26
3. 17
4. 273
5. 171
6. 16
7. Letters

**BCPI/Data Storage**  
**Unit 3: Data Representation and Numbering Systems**  
**Teacher's Guide/ANSWERS**

**Part VII**

**Project 1**

|    | Binary | Octal | Hexidecimal |
|----|--------|-------|-------------|
| 1. | 2000   | 20    | 10          |
| 2. | 3201   | 41    | 21          |
| 3. | 1000   | 10    | 8           |
| 4. | 1011   | 12    | B           |
| 5. | 10000  | 100   | 40          |

**Project 2**

|     |   |     |   |
|-----|---|-----|---|
| 1.  | B | 14. | B |
| 2.  | D | 15. | A |
| 3.  | C | 16. | A |
| 4.  | D | 17. | B |
| 5.  | C | 18. | A |
| 6.  | A | 19. | B |
| 7.  | C | 20. | A |
| 8.  | A | 21. | B |
| 9.  | D | 22. | A |
| 10. | B | 23. | A |
| 11. | B | 24. | A |
| 12. | B | 25. | A |
| 13. | C |     |   |

## COMPETENCY CHECKLIST

### ESSENTIAL ELEMENT: 75.87(g)(6)

Students will be able to demonstrate knowledge of computer numbering systems and data representation on storage devices.

| OBJECTIVE: | Students will be able to demonstrate knowledge of computer numbering systems. |         |             |
|------------|-------------------------------------------------------------------------------|---------|-------------|
|            |                                                                               | MASTERY | NON-MASTERY |
| 1.         | Define character and control codes.                                           | _____   | _____       |
| 2.         | Give examples of ASCII coding.                                                | _____   | _____       |
| 3.         | Give examples of EBCDIC coding.                                               | _____   | _____       |
| 4.         | Demonstrate an understanding of the binary numbering system.                  | _____   | _____       |
| 5.         | Demonstrate an understanding of the octal numbering system.                   | _____   | _____       |
| 6.         | Demonstrate an understanding of the hexadecimal numbering system.             | _____   | _____       |

STUDENT NAME \_\_\_\_\_

## SECTION IV: DATA STORAGE

### UNIT I: MEDIA AND TYPES OF STORAGE

#### UNIT

**OBJECTIVE:** The students will be able to identify primary and secondary storage devices and media.

#### STUDENT

**COMPETENCIES:** After completing this unit, students will be able to:

1. Define primary and secondary storage.
2. Identify secondary storage devices and media.
3. Distinguish between volatile and nonvolatile memory.
4. Differentiate different file organizations and file accesses.

#### FOCUS:

To process and store data, one must understand the role of primary and secondary storage. Application requirements will determine the type of secondary storage and method of access.



## TERMINOLOGY

|                         |                                                                                                                                             |
|-------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Auxiliary</b>        | A supplemental storage area outside of primary storage. Also referred to as external or secondary storage.                                  |
| <b>Batch processing</b> | A form of processing information that collects many transactions before a group is processed.                                               |
| <b>Bit</b>              | Abbreviation of binary digit. A bit can be 0 or 1.                                                                                          |
| <b>Bubble memory</b>    | A recent auxiliary storage device in which data is stored by magnetized spots or bubbles on a thin film of semiconductor material.          |
| <b>Buffer</b>           | A storage device where data is stored to compensate for the differences in speed rates between devices.                                     |
| <b>Byte</b>             | Usually an eight-bit configuration where one character can be stored.                                                                       |
| <b>Cache</b>            | A high-speed buffer located in the CPU to speed processing.                                                                                 |
| <b>Channel</b>          | A track located on magnetic tape. A channel can also be a band on a magnetic drum.                                                          |
| <b>Chip</b>             | A thin wafer made of silicon where the integrated electronic components are imprinted.                                                      |
| <b>CPU</b>              | Central Processing Unit. The component of a computer system that includes primary storage, arithmetic-logic unit, and control unit.         |
| <b>Cylinder</b>         | All tracks of the same number on different magnetic disks make up a cylinder.                                                               |
| <b>DASD</b>             | Direct-access storage device that allows data to be accessed or stored either sequentially or randomly.                                     |
| <b>Direct access</b>    | A method whereby data can be accessed directly, depending upon the physical location of the information. Also referred to as random access. |
| <b>Disk</b>             | A spinning platter where data and programs can be stored by the use of magnetized spots.                                                    |
| <b>Disk drive</b>       | The device where the disk is placed to be on-line to the computer. It is considered to be the most popular form of auxiliary storage.       |

|                      |                                                                                                                                                                                 |
|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Disk pack</b>     | A stack of disks on a spindle. Information can be directly accessed when mounted on the disk drive. A popular secondary storage medium in large computer installations.         |
| <b>Diskette</b>      | Commonly referred to as a floppy disk or flexible disk. A popular secondary storage medium for small and medium computer installations.                                         |
| <b>Hard disk</b>     | A disk that does not bend.                                                                                                                                                      |
| <b>Internal</b>      | Primary storage that is addressable in the storage CPU.                                                                                                                         |
| <b>K</b>             | Represents 1024 storage units, or bytes, when referring to main memory.                                                                                                         |
| <b>Keypunch</b>      | A machine used to punch holes in a card to represent data.                                                                                                                      |
| <b>Key-to-disk</b>   | A machine that enters data directly to a magnetic disk or diskette.                                                                                                             |
| <b>Key-to-tape</b>   | A machine that enters data directly to a magnetic tape.                                                                                                                         |
| <b>Laser disk</b>    | A secondary storage device in which information is recorded by holes burned onto the surface of the disk by a laser beam.                                                       |
| <b>LSI</b>           | Large Scale Integration. A single silicon chip that is packed with circuits containing thousands of transistors.                                                                |
| <b>Magnetic core</b> | A small doughnut-shaped ring that has wires passing through. The flowing of electrical current will magnetize or demagnetize the core representing the "on" or "off" condition. |
| <b>Magnetic disk</b> | An auxiliary storage medium consisting of a metal platter in which magnetized spots can be used to record data. Disks can be hard or floppy.                                    |
| <b>Magnetic drum</b> | Data can be stored on a magnetic cylinder. Drums hold vast amounts of data.                                                                                                     |
| <b>Magnetic tape</b> | A storage medium consisting of a long strip of flexible plastic that can be magnetized to store information.                                                                    |
| <b>Main storage</b>  | The memory located in the CPU.                                                                                                                                                  |
| <b>Media</b>         | The material where data is recorded or stored.                                                                                                                                  |
| <b>Medium</b>        | The singular of media.                                                                                                                                                          |
| <b>Megabyte</b>      | One million bytes of storage.                                                                                                                                                   |

|                           |                                                                                                            |
|---------------------------|------------------------------------------------------------------------------------------------------------|
| <b>Memory</b>             | Storage locations whereby information can be accessed or stored.                                           |
| <b>Nanosecond</b>         | One-billionth of a second.                                                                                 |
| <b>Nibble</b>             | One-half of a byte; four bits.                                                                             |
| <b>Nonvolatile</b>        | Data is not lost when the power is turned off.                                                             |
| <b>Numeric address</b>    | A specific location in memory.                                                                             |
| <b>Off-line</b>           | Hardware is not directly connected to the computer.                                                        |
| <b>On-line</b>            | Connected to the computer.                                                                                 |
| <b>Peripherals</b>        | Includes any hardware other than the CPU. Examples: I/O devices and auxiliary storage devices.             |
| <b>Picosecond</b>         | One-thousandth of a nanosecond.                                                                            |
| <b>Primary</b>            | Commonly called main memory or internal storage located in the CPU.                                        |
| <b>RAM</b>                | Random-Access Memory. Memory available to be used for processing or storage.                               |
| <b>Random access</b>      | A method of accessing data directly.                                                                       |
| <b>Register</b>           | An internal part of the computer that can store and transfer an instruction or data.                       |
| <b>ROM</b>                | Read-Only Memory. The section of the computer's memory where information cannot be deleted or changed.     |
| <b>Secondary storage</b>  | Same as auxiliary storage.                                                                                 |
| <b>Sector</b>             | A segment of a disk.                                                                                       |
| <b>Sequential</b>         | A method of accessing storage whereby data must be read in sequence in order to access the desired record. |
| <b>Sequential storage</b> | Records that are stored one after another in a file, in a specific order.                                  |
| <b>Spindle</b>            | The vertical shaft of a hard disk drive where several hard disks are stacked.                              |
| <b>Storage</b>            | Describes a device or medium that can save and retrieve data.                                              |

**BCPI/Data Storage**  
**Unit 1: Media and Types of Storage**  
**Student's Guide/TERMINOLOGY**

|                 |                                                                                                                                                                            |
|-----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Track</b>    | A horizontal path, the length of a magnetic tape, on which data can be recorded magnetically; one concentric circle on a disk or diskette, or one circular band on a drum. |
| <b>Virtual</b>  | A method whereby secondary storage is utilized giving the impression that primary storage is unlimited.                                                                    |
| <b>Volatile</b> | That part of memory that loses data when the power is turned off.                                                                                                          |

628

## UNIT OVERVIEW

- I. A computer's storage or memory can be defined as either primary or secondary.
  - A. The first microcomputers had only 12K (thousand) bytes of primary memory which was equivalent to some of the small mainframe computers.
  - B. Today's microcomputers come with a minimum of 256K, and mainframe computers measure memory in terms of megabytes (million) of memory.
  - C. In like comparison, the speed of processing has made tremendous strides in being able to process information in the main memory.
- II. Primary memory (also called main storage) consists of memory chips located in the CPU (central processing unit) which vary in the amount of information it can store.
  - A. Memory chips or processing boards can be added to most computers to increase memory or processing.
  - B. Most BASIC interpreters and some operating systems can be stored in primary memory (usually referred to as ROM/read-only memory) at the factory.
  - C. RAM (random-access memory) is the primary memory available for processing by the programmer or operator.
    1. RAM is volatile.
    2. The information in memory is lost when the power is turned off.
    3. Information in RAM must be saved to a secondary storage medium for future processing.
  - D. ROM is read-only memory and cannot be changed or altered by the operator.
    1. ROM is nonvolatile.
    2. The information in memory is not lost when the power is turned off.
- III. Secondary memory (also called auxiliary storage) includes various forms of storage other than primary memory.
  - A. The most common forms of secondary storage are magnetic tape and disk (hard or floppy).

- B. Other forms of secondary storage are: punch cards, paper tape, magnetic drums, magnetic bubble storage, charge coupled devices, mass storage devices, and optical disk technology.
- C. When these media are being used on the computer, they are said to be on-line to the computer. When they are not being used on the computer, they are off-line.
- D. Information can be stored or accessed serially (sequential) or randomly (direct).
  - 1. Recording or reading each record in sequence is the serial or sequential method.
    - a. This is the method used in recording information on magnetic tape.
    - b. Thus, record 99 can be read only by reading the first 98 to get to record 99.
  - 2. Information on disk can be accessed direct; although, it may have been recorded sequentially. Direct access allows one to go directly to the record requested.
    - a. Magnetic tape can be more economical, but processing is slower than with magnetic disks. Processing speed becomes extremely important when working with large files.
    - b. Most businesses use direct access media, which include magnetic disks, magnetic drums, mass storage devices, magnetic bubble storage, charge coupled devices, and optical disk technology.

## ACTIVITY 1: PRIMARY AND SECONDARY STORAGE

**INSTRUCTIONS:** Complete the following sentences, and then define primary and secondary storage.

1. Primary storage is also called \_\_\_\_\_ storage.
2. Secondary storage is also called \_\_\_\_\_ storage.
3. Primary Storage:
4. Secondary Storage:

## ACTIVITY 2: SECONDARY STORAGE DEVICES AND MEDIA

**INSTRUCTIONS:** Match each of the characteristics of secondary storage media with the name of the medium. (Some characteristics may apply to more than one medium.) Write the letters corresponding to the medium in the space provided.

### MEDIA

- |    |               |    |               |
|----|---------------|----|---------------|
| A. | Magnetic tape | C. | Mass storage  |
| B. | Magnetic disk | D. | Bubble memory |

### CHARACTERISTICS

- \_\_\_\_\_ 1. Mounted on a spindle.
- \_\_\_\_\_ 2. Newest type of secondary storage.
- \_\_\_\_\_ 3. Data recorded in horizontal rows (channels).
- \_\_\_\_\_ 4. Data stored in cartridges in data cells.
- \_\_\_\_\_ 5. Data recorded in tracks.
- \_\_\_\_\_ 6. Allows random access of data.
- \_\_\_\_\_ 7. Data stored and retrieved sequentially only.
- \_\_\_\_\_ 8. Data organized according to cylinders or sectors.



### ACTIVITY 3: VOLATILE AND NONVOLATILE MEMORY

#### Project 1

**INSTRUCTIONS:** Distinguish between volatile and nonvolatile memory.

Volatile memory

Nonvolatile memory

#### Project 2

**INSTRUCTIONS:** Fill in the correct answers to complete the statements below.

1. ROM stands for \_\_\_\_\_.
2. The memory in ROM is \_\_\_\_\_.
3. RAM stands for \_\_\_\_\_.
4. The memory in RAM is \_\_\_\_\_.
5. The memory available to the programmer is referred to as \_\_\_\_\_.

## ACTIVITY 4: FILE ORGANIZATION AND ACCESS METHODS

### Project 1

**INSTRUCTIONS:** Compare the differences between random-and sequential-file organization.

Random:

Sequential:

### Project 2

**INSTRUCTIONS:** Define serial and random-file access.

Serial:

Random:

## SECTION IV: DATA STORAGE

### UNIT II: DATA STORAGE APPLICATIONS

#### UNIT

#### OBJECTIVE:

Students will be able to write a program to create and maintain a data file.

#### STUDENT

#### OBJECTIVES:

After completing this unit, the student will be able to:

1. Identify data elements--field, record, and file.
2. Code a program to create a sequential and/or random data file.
3. Code a program to retrieve data and produce a report from a data file they created.

#### FOCUS:

When data are to be saved, programs must be developed to store and access the data file.

## TERMINOLOGY

|                      |                                                                                                                                                           |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Access</b>        | Locating specific data. Examples: direct access, random access, and serial access.                                                                        |
| <b>Access time</b>   | The time between requesting data and the delivery of data from storage.                                                                                   |
| <b>Address</b>       | The address of a register location that contains a data item.                                                                                             |
| <b>Data base</b>     | A group of data needed by a business or individuals to process information.                                                                               |
| <b>Field</b>         | A unit of information such as name, address, or city.                                                                                                     |
| <b>File</b>          | A group or unit of records.                                                                                                                               |
| <b>File access</b>   | The method used in storing and retrieving data on auxiliary storage.                                                                                      |
| <b>Indexed</b>       | An organization of records that are placed in a file sequentially, and an index is maintained whereby the file can be accessed sequentially and directly. |
| <b>Indexed file</b>  | A file in which each record is found in one or more indexes for storage and retrieval purposes.                                                           |
| <b>ISAM</b>          | Index Sequential Access Method. A method whereby sequential records can be directly accessed by use of an index based on some key field.                  |
| <b>Key</b>           | A unique item of a record that is used in locating a specific record.                                                                                     |
| <b>Primary key</b>   | A unique field of a record that is used to locate a specific record.                                                                                      |
| <b>Random file</b>   | A file where data can be stored and retrieved directly.                                                                                                   |
| <b>Record</b>        | A collection of related fields.                                                                                                                           |
| <b>Relative file</b> | A random file where a record can be stored or retrieved by using its record number.                                                                       |
| <b>Secondary key</b> | Fields that can be used to access records, but which are not unique.                                                                                      |

## UNIT OVERVIEW

- I. The physical organization of information stored on secondary storage consists of files, records, and fields.
  - A. A file can be defined as a collection or group of records.
  - B. A record can be defined as a unit of information.
  - C. A field is a subdivision within the unit of information.
  - D. Thus, a file would contain all the students in the school; a record would be information concerning one student; and a field would be the student name, ID number, etc.
- II. Database programs, such as Lotus 1-2-3 or dBase III, provide an easy way to create and manipulate data files without any programming knowledge. Larger computer systems call such programs "utilities."
- III. Special processing requires that a programmer write a program in a high-level language that will create and access a data file. A great deal of planning and many decisions must be made to set up a file. Some of the decisions that need to be made are:
  - A. Which type of file would be most appropriate.
  - B. The amount of data, how the information will be used, who will be using it, what the necessary speed requirements should be, and what type of reports will be needed.
  - C. Which way to organize files--sequential, direct, relative, or indexed.
    1. The records in sequential files are recorded one after another and are retrieved in the same manner.
    2. Direct or random files may be stored or retrieved in any particular order by specifying the disk address. The disk address includes the track and sector number.
    3. Records in a relative file can be accessed by the relative position in the file normally associated with the key field, which determines the relative position of the file.
      - a. A computer program can be written that will convert key fields to record numbers.
      - b. Records can be accessed rapidly using relative files.

4. An indexed file is necessary when getting the information requires accessing by more than one field.
  - a. Indexed files can be compared to an index in a book. Records can be accessed directly by looking up information in the appropriate index, whether it is by name, account number, etc.
  - b. Indexed files offer the advantage of providing access to files by different fields, but having to search different indexes slows access time.

## ACTIVITY 1: DEFINE DATA ELEMENTS

**INSTRUCTIONS:** Define a file, a record, and a field.

File:

Record:

Field:

## ACTIVITY 2: CREATING SEQUENTIAL AND RANDOM FILES

### Project 1

**INSTRUCTIONS:** Write a BASIC program to create a sequential data file containing the fields: name, address, city, state, and zip. Name the data file *addxx*, and save on your data disk. Name the program *seqlxx* (*xx* is your initials). Submit program listing, coding, and data file list to your instructor.

### Project 2

**INSTRUCTIONS:** Write a program to access the data file (*addxx*) created in Project 1. The headings should include your name, File Display - ADDxx, and an appropriate main heading. The detail line should include the name, address, city, state, and zip. Name the program *aaddxx*. Use your judgment in preparing a printer spacing chart. Submit output, program listing, coding, and printer spacing chart to your instructor.

**Input:**

| Name            | Address          | City      | State | Zip   |
|-----------------|------------------|-----------|-------|-------|
| Jones, Ray      | 40198 Fallas Rd  | Ft. Worth | TX    | 76101 |
| Kelley, Sam     | 3344 Longview    | Dallas    | TX    | 77801 |
| Lowell, Martha  | 19098 W. Cedar   | Houston   | TX    | 73203 |
| Norris, Roger   | 258 Hillside St. | Austin    | TX    | 70303 |
| Overman, Robert | 3877 Round Ave.  | Dallas    | TX    | 77803 |
| Young, Willis   | 3244 Cherry Lane | Ft. Worth | TX    | 76116 |



## ACTIVITY 2 (Continued)

### Project 3

**INSTRUCTIONS:** Write a BASIC program to create a sequential file containing the fields: item number, quantity on hand, description, quantity on order, unit measure, unit price and unit cost. Name the data file *invxx* and save on your data disk. Name the program *seq2xx*. Submit program listing, coding, and data file list to your instructor.

### Project 4

**INSTRUCTIONS:** Write a program to access the data file (*invxx*) created in Project 3. The headings should include your name, File Display - INVxx, and an appropriate main heading. The detail line should include: on order quantity, item number, description, unit price, and the total price (on order quantity X unit cost). Name the program *ainvxx*. Use your judgment in preparing a printer spacing chart. Submit output, program listing, coding, and printer spacing chart to your instructor.

**Input:**

| Item # | OH Quan | Descrip.      | OO Quan | U/M | Price | Cost  |
|--------|---------|---------------|---------|-----|-------|-------|
| 63219  | 1500    | Dinner Plate  | 500     | ea  | 4.75  | 2.52  |
| 45011  | 350     | Salad Plate   | 400     | ea  | 12.95 | 7.50  |
| 1205   | 414     | Serving Plate | 5       | ea  | 16.50 | 8.25  |
| 1317   | 756     | Small Bowl    | 0       | ea  | 8.22  | 4.72  |
| 3298   | 710     | Silver Spoons | 5       | dz  | 35.68 | 19.95 |
| 8923   | 330     | Blue Crystal  | 20      | ea  | 9.25  | 5.45  |

## ACTIVITY 2 (Continued)

### Project 5

**INSTRUCTIONS:** Write a BASIC program to create a direct data file containing the fields: social security number, name, regular hours, rate of pay, deductions. Name the data file *payxx* on your data disk. Name the program *dirlxx*. Submit program listing and coding to your instructor.

### Project 6

**INSTRUCTIONS:** Write a program to access the data file created in Project 5 by record number. Display social security number, name, regular hours, and total pay (regular hours X rate - deductions). Name the program *apayxx*. Your instructor may vary the desired output. Submit output, program listing, coding, and printer spacing chart to your instructor.

**Input:**

| SS #      | Name          | Regular Hrs. | Rate | Deductions |
|-----------|---------------|--------------|------|------------|
| 814191632 | Wilson Wycoff | 40           | \$5  | 10         |
| 210014332 | Joe Manley    | 35           | \$4  | 12         |
| 005099428 | John Swan     | 15           | \$4  | 4          |
| 410152102 | Amy Rayburn   | 42           | \$5  | 17         |
| 854385247 | Jeff Davis    | 40           | \$8  | 22         |
| 312403880 | Gladys Benton | 33           | \$5  | 12         |

### ACTIVITY 3: RETRIEVE DATA

#### Project 1

**INSTRUCTIONS:** Write a BASIC program to create a direct data file containing the fields: quantity, description, product number, unit price. Name the data file *salexx* on your data disk. Name the program *dir2xx*. Submit program listing and coding to your instructor.

#### Project 2

**INSTRUCTIONS:** Write a program to access the data file created in Project 1 by record number. Display quantity, unit price, description, and total price (quantity x unit price). Name the program *asalexx*. Your instructor may vary the desired output. Submit output, program listing, coding, and printer spacing chart to your instructor.

**Input:**

| Quantity | Description   | Product # | Unit Price |
|----------|---------------|-----------|------------|
| 61       | Salt          | 43781     | 7.30       |
| 66       | Crackers      | 48312     | .68        |
| 67       | Peanut Butter | 27649     | 1.55       |
| 71       | Coffee        | 45263     | 2.88       |
| 77       | Corn          | 23289     | 7.70       |
| 145      | Flour         | 11383     | 2.75       |

## SECTION IV: DATA STORAGE

### UNIT III: DATA REPRESENTATION AND NUMBERING SYSTEMS

#### UNIT

**OBJECTIVE:** Students will be able to demonstrate knowledge of computer numbering systems.

#### STUDENT

**OBJECTIVES:** After completing this unit, the student will be able to:

1. Define character and control codes.
2. Give examples of ASCII coding.
3. Give examples of EBCDIC coding.
4. Demonstrate an understanding of the binary numbering system.
5. Demonstrate an understanding of the octal numbering system.
6. Demonstrate an understanding of the hexadecimal numbering system.

**FOCUS:** In order to create and manipulate data files, one should understand data representation as it relates to primary and secondary storage.

## TERMINOLOGY

|                          |                                                                                                                                                                                                              |
|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ASCII                    | American Standard Code for Information Interchange. A standard, seven-bit code used for representing data in a computer system and information interchange between computer systems.                         |
| Base                     | A number that is multiplied when it is moved a position to the left.                                                                                                                                         |
| BCD                      | Binary-Coded Decimal                                                                                                                                                                                         |
| Binary number            | The character 0 or 1 with a base of 2. Zero system representing "off", and one representing "on."                                                                                                            |
| Character                | Any character that can be represented using data. The internal code of the computer.                                                                                                                         |
| Control                  | A control character performs an operation which does not appear as a character on the screen. Examples: backspace, center, tab, etc. The control character is sending a command to the computer or a device. |
| Decimal numbering system | A numbering system using the base of 10.                                                                                                                                                                     |
| EBCDIC                   | Extended Binary-Coded Decimal Interchange Code. A popular eight-bit code used for representing data in a computer system.                                                                                    |
| Hexadecimal              | A base-16 numbering system using the digits 0-9 and characters A-F.                                                                                                                                          |
| Octal number             | A numbering system using base-8 system.                                                                                                                                                                      |
| Parity (check)           | Used to check the accuracy of data bit transmission. It is an internal checking system that counts the number of bits turned on to be odd or even.                                                           |
| Radix                    | Synonymous with base.                                                                                                                                                                                        |
| VLSI                     | Very Large Scale Integration.                                                                                                                                                                                |

## UNIT OVERVIEW

- I. Most small computers use ASCII (American Standard Code for Information Interchange) code as the internal numeric code accepted by the computer.
  - A. ASCII uses the numbers 0-128 and some expanded systems 0-256 numbers.
  - B. The numbers 0-31 are called the nonprinting, or control, characters. These codes have to do with the operation of the machine and therefore are not produced on the screen.
  - C. The numbers 32-127 are the character codes which produce characters on the screen.
  - D. ASCII is a widely used code for data communications between computers and devices. It is critical that computers have the same internal code when communicating, or no communication will take place.
- II. Larger computer systems use EBCDIC (Extended Binary-Coded Decimal Interchange Code) as the internal numbering system.
  - A. EBCDIC uses the numbers 0-256.
  - B. A translation must be made between ASCII and EBCDIC computers in order for them to be able to communicate.
- III. *Binary* means two. Thus two conditions exist, 1 or 0.
  - A. The circuitry within the computer is quite simple; either a bit (binary digit) is turned on--"1", or it is turned to off--"0". Electricity is either present or absent; that is, magnetized or not magnetized.
  - B. Traditionally, each position in a numbering system is referred to as a bit.
    1. Eight bits make up a byte. The letter R is 82 in ASCII.
    2. The binary equivalent would be:

### Example 1

|          |     |    |    |    |   |   |   |   |                           |
|----------|-----|----|----|----|---|---|---|---|---------------------------|
|          | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |                           |
| binary = | 1   | 0  | 1  | 0  | 0 | 1 | 0 |   | ASCII= 82<br>Character= R |

### Example 2

|  |     |    |    |    |   |   |   |   |                           |
|--|-----|----|----|----|---|---|---|---|---------------------------|
|  | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |                           |
|  | 1   | 1  | 1  | 0  | 0 | 1 |   |   | ASCII= 57<br>Character= 9 |

- C. One-half of a byte is referred to as a *nibble* (four bits).
  - D. The EBCDIC or ASCII code is converted to a binary number which exists of 1's or 0's.
  - E. In our everyday life, we are familiar with the decimal system which uses a base of 10. There is the 1 position, 10 position, 100 position, 1000 position, etc.
  - F. The binary numbering system uses a base of 2. There is the 1 position, 2 position, 4 position, 8 position, etc. Each position is multiplied by two.
  - G. The octal numbering system uses a base of 8. The beginning number would be 1 position, 8 position, 64 position, 512 position, etc. Each position is multiplied eight.
  - H. The hexadecimal numbering system uses a base of 16. The beginning number would be 1 position, 16 position, 256 position, etc. Every numbering system uses the digits 1-9. The hexadecimal uses the letters A-F also. The letter A = 10, B = 11, C = 12, D = 13, E = 14, F = 15; thus 16 characters are being used.
  - I. It must be kept in mind that the computer reads a bit from right to left.
- IV. An additional bit in the byte configuration is the parity bit. The purpose of the parity bit is to check whether the number of bits per byte is odd or even.
- A. When data is transmitted internally, a self-checking feature is built into the computer to detect if bits are "dropped," or turned off, while the operation is occurring.
  - B. A computer is set up at the factory to check for an odd or even number of bits turned on. If the opposite condition exists, then the operator will get a parity error message. Refer to example 1 above and note that three bits are turned on. A ninth bit would be added to the high-order position (left-most) as a parity bit.
  - C. If a computer checks for even parity, the parity bit will automatically be turned on by the computer. If a bit is dropped during transmission, a parity error message will be displayed to indicate a problem. In example 2, the parity bit will remain off since four bits turned on are an even number.
  - D. Even though it is not critical for the programmer to have an in depth knowledge of the internal coding, it does help in the understanding of how a computer operates and helps in the diagnosis of errors.
  - E. When programming in machine language, or assembler, the understanding of the code becomes more critical.



## ACTIVITY 1: CONTROL CODES

**INSTRUCTIONS:** Indicate whether each of the following statements is (T) rue or (F) alse.

### U.S.A. STANDARD CODE (ASCII)

- \_\_\_\_\_ 1. Allows computers made by different manufacturers to be linked together.
- \_\_\_\_\_ 2. Zone and digit configuration are the same as EBCDIC.
- \_\_\_\_\_ 3. Developed for users of telecommunications networks.
- \_\_\_\_\_ 4. Developed in 1967.
- \_\_\_\_\_ 5. An extended code allows up to 256 characters.

### EXTENDED BINARY-CODED DECIMAL INTERCHANGE CODE (EBCDIC)

- \_\_\_\_\_ 6. Used in second-generation computers.
- \_\_\_\_\_ 7. Used in third- and fourth-generation computers.
- \_\_\_\_\_ 8. Can represent up to 256 characters.
- \_\_\_\_\_ 9. Can represent up to 64 characters.
- \_\_\_\_\_ 10. Includes both upper- and lower-case characters.

### BINARY-CODED DECIMAL SYSTEM (BCD)

- \_\_\_\_\_ 11. Used in second-generation computers.
- \_\_\_\_\_ 12. Used in third- and fourth-generation computers.
- \_\_\_\_\_ 13. Can represent over 64 characters.
- \_\_\_\_\_ 14. Can represent up to 64 characters.
- \_\_\_\_\_ 15. Based on the binary code.



## ACTIVITY 2: USING A CODE AND REVIEW OF RELATED TERMS

### Project 1

**INSTRUCTIONS:** Use the following telephone code to translate each of the digits into data processing terms or acronyms.

|          |          |          |
|----------|----------|----------|
| 1        | ABC<br>2 | DEF<br>3 |
| GHI<br>4 | JKL<br>5 | MNO<br>6 |
| PRS<br>7 | TUV<br>8 | WXY<br>9 |
| *        | 0        | #        |

1. 27244 \_\_\_\_\_
2. 248 \_\_\_\_\_
3. 246279 \_\_\_\_\_
4. 727489 \_\_\_\_\_
5. 24325 248 \_\_\_\_\_
6. 223 \_\_\_\_\_
7. 322342 \_\_\_\_\_
8. 2983 \_\_\_\_\_
9. 43923324625 \_\_\_\_\_
10. 62825 \_\_\_\_\_

649

## ACTIVITY 2 (Continued)

### Project 2

**INSTRUCTIONS:** Match each term in Project 1 with the correct definition below by writing the corresponding number in the space provided.

- |       |     |                                                                                  |
|-------|-----|----------------------------------------------------------------------------------|
| _____ | 1.  | Smallest unit of data that can be recognized by the computer.                    |
| _____ | 2.  | Number system using digits 0-9 and letters A-F, base 16.                         |
| _____ | 3.  | Number system using digits 0 and 1, base 2.                                      |
| _____ | 4.  | Binary digit used to check accuracy.                                             |
| _____ | 5.  | An error condition during transmission.                                          |
| _____ | 6.  | Smallest unit of data that can be addressed within a computer.                   |
| _____ | 7.  | Number system using digits 0-7, base 8.                                          |
| _____ | 8.  | Standard code used in teleprocessing.                                            |
| _____ | 9.  | Eight-bit code for character representation; can represent up to 256 characters. |
| _____ | 10. | Six-bit code for data representation; can represent up to 64 characters.         |

### ACTIVITY 3: BINARY AND DECIMAL CONVERSION

#### Project 1

**INSTRUCTIONS:** Convert the following binary numbers to decimal numbers.

**Example:** Binary - 00100010  
Decimal - 34 (32 + 2)

- |    |                   |                 |
|----|-------------------|-----------------|
| 1. | Binary - 1101     | Decimal - _____ |
| 2. | Binary - 10010001 | Decimal - _____ |
| 3. | Binary - 0110     | Decimal - _____ |
| 4. | Binary - 01100100 | Decimal - _____ |

#### Project 2

**INSTRUCTIONS:** Convert the following decimal numbers to binary numbers.

- |    |               |                |
|----|---------------|----------------|
| 1. | Decimal - 25  | Binary - _____ |
| 2. | Decimal - 50  | Binary - _____ |
| 3. | Decimal - 12  | Binary - _____ |
| 4. | Decimal - 130 | Binary - _____ |

## ACTIVITY 4: OCTAL AND DECIMAL CONVERSION

### Project 1

**INSTRUCTIONS:** Convert the following octal numbers to decimal numbers.

**Example:** Octal - 17  
Decimal - 15 ( $8 + 7$ )

- |    |              |                 |
|----|--------------|-----------------|
| 1. | Octal - 0012 | Decimal - _____ |
| 2. | Octal - 0075 | Decimal - _____ |
| 3. | Octal - 0200 | Decimal - _____ |
| 4. | Octal - 1000 | Decimal - _____ |

### Project 2

**INSTRUCTIONS:** Convert the following decimal numbers to octal numbers.

**Example:** Decimal - 20  
Octal - 0024 ( $16 + 4 = 20$ )

- |    |               |               |
|----|---------------|---------------|
| 1. | Decimal - 6   | Octal - _____ |
| 2. | Decimal - 12  | Octal - _____ |
| 3. | Decimal - 50  | Octal - _____ |
| 4. | Decimal - 100 | Octal - _____ |

## ACTIVITY 5: HEXADECIMAL AND DECIMAL CONVERSION

### Project 1

INSTRUCTIONS: Convert the following hexadecimal numbers to decimal numbers.

Example:   Hexadecimal - AE  
              Decimal - 174 (160 + 14)

- |    |                  |                 |
|----|------------------|-----------------|
| 1. | Hexadecimal - F  | Decimal - _____ |
| 2. | Hexadecimal - 5  | Decimal - _____ |
| 3. | Hexadecimal - 23 | Decimal - _____ |
| 4. | Hexadecimal - 32 | Decimal - _____ |

### Project 2

INSTRUCTIONS: Convert the following decimal numbers to hexadecimal numbers.

Example:   Decimal - 25  
              Hexadecimal - 19

- |    |               |                     |
|----|---------------|---------------------|
| 1. | Decimal - 120 | Hexadecimal - _____ |
| 2. | Decimal - 12  | Hexadecimal - _____ |
| 3. | Decimal - 256 | Hexadecimal - _____ |
| 4. | Decimal - 7   | Hexadecimal - _____ |

## **CLOSURE OF BUSINESS COMPUTER PROGRAMMING**

This section provides a closure to the course Business Computer Programming. The section includes review of course objectives, sections, units, and student competencies. Suggestions for course and student evaluations are also included.

## SUGGESTED UNIT LESSON PLAN

### CLOSURE FOR BUSINESS COMPUTER PROGRAMMING

**ALLOCATED  
TIME: 3 Hours**

**UNIT**

**OBJECTIVE:** Students will be able to analyze, evaluate, and demonstrate proficiency in course content.

**STUDENT**

**COMPETENCIES:** After completing this unit, the students will be able to:

1. Relate the objectives of the course Business Computer Programming to personal success in the business environment.
2. Analyze section and unit contents in relation to the course objectives.
3. Recognize levels of competency performance.
4. Assess the course in terms of objectives and content.
5. Complete a course examination with a score of 70 percent or better accuracy.

**FOCUS:**

An important component of the learning process is closure. Closure may be considered a review, a summation, or a conclusion. The student is led to see how the parts make up or contribute to the whole. A final question needs to be answered: "Has the course accomplished its primary objective?"

**DOMAINS:**

Activities and competency tests have been designed to provide learning experiences utilizing the cognitive, psychomotor, or affective domains. Activities and competency tests are coded on the Activity and Competency Test Matrices.

**BASIC SKILLS:**

Activities and competency tests have been designed to reinforce the basic skills of reading, writing, and/or mathematics. Activities and competency tests have been coded on the Activity and Competency Test Matrices.

## CONTENT OUTLINE

- I. Review of course objectives
- II. Review of sections and units
  - A. Introduction to Business Computer Programming
  - B. Computer hardware utilization
  - C. Program development
  - D. Data storage
- III. Analysis of student competencies
- IV. Course evaluation
- V. Examination

656



## SUGGESTED TEACHER ACTIVITIES

### General Teacher Preparation

1. Preview unit.
2. Review unit objectives.
3. Make instructional analysis.
4. Assess student learning needs.
5. Decide on lesson domain: cognitive, psychomotor, or affective domain.
6. Select instructional materials based on instructional analysis and student learning needs.
7. Select instructional strategies based on lesson content and student learning needs.
8. Prepare lesson plans from the unit lesson.
9. Prepare additional instructional material.
10. Reserve necessary media and equipment.

### Teacher Presentation

1. Present student objectives.
2. Establish rapport with students.
3. Review lesson objectives with students.
4. Relate lesson to prior or future learning.
5. When appropriate, encourage student interaction through questioning, discussions, or some other strategy.
6. Emphasize the value/importance of the unit or activity.
7. Allow time for closure of a lesson.

### Reteach Activities

1. Identify area(s) of weakness from results of evaluation.
2. Select unit activities not previously used.
3. Use additional resources and/or create materials based on student learning needs.
4. Use a different teaching strategy in reteaching.
5. Retest.

Specific Teaching Activities

1. Design a bulletin board for Business Computer Programming I. Suggestions are: WHAT I HAVE LEARNED FROM BCPI; MY BEST FROM BCPI; AFTER BCPI, WHAT?
2. Review overhead transparencies included with Introduction unit on course objectives.
3. Discuss with students the objectives of the course.
4. Divide students into groups. Assign to each group a section to be reviewed and presented to the class. The review might include: competencies, synopsis of units, importance of units. Have the other students take notes.
5. Hand out the listing of competencies to the class. Have each student rate what he/she considers to be his/her level of performance for each competency.

Suggested levels of performance might include: MIN (minimum), 1; F (fair), 2; A (average), 3; or M (mastery), 4. The instructor will need to determine what criteria constitute the levels. Collect these sheets and tally the information for each of the competencies. Use this information when planning for the next year. Make revisions in your teaching style, presentation, etc., for competencies that a majority of the students feel they have achieved only a minimal mastery.

6. Assign students to develop review questions for the examination. Provide the students with the procedures for keying in the questions. Make copies for each group of questions and answers for the remaining class members. Review these with the students. Inform the students that you will be adding some questions of your own. Provide the students with examples. Give the examination from these questions.
7. Give examples for the type of psychomotor skills you will be testing.
8. Assign time for students to review for their examination. Be sure they know the correct answers to the questions and/or activities.

## CLOSURE FOR BUSINESS COMPUTER PROGRAMMING

### UNIT

**OBJECTIVE:** Students will be able to analyze, evaluate, and demonstrate proficiency in course content.

### STUDENT

**COMPETENCIES:** After completing this unit, the student will be able to:

1. Relate the objectives of the course Business Computer Programming to personal success in the business environment.
2. Analyze section and unit contents in relation to the course objectives.
3. Recognize levels of competency performance.
4. Assess the course in terms of objectives and content.
5. Complete a course examination with a score of 70 percent or better accuracy.

### FOCUS:

An important component of the learning process is closure. Closure may be considered a review, a summation, or a conclusion. The student is led to see how the parts make up or contribute to the whole. A final question needs to be answered: "Has the course accomplished its primary objective?"

## UNIT OVERVIEW

- I. Review of course objectives
- II. Review of sections and units
  - A. Introduction to Business Computer Programming
    1. Introduction to the course
    2. Social implications
      - a. Identify terminology related to computer crime, computer users, and new computer devices and techniques.
      - b. Identify the major issues and technological developments related to computers and society.
      - c. Describe the ethical and legal issues that have arisen as a result of the computer revolution.
      - d. Identify privacy issues involved with electronic data storage.
      - e. Explain the reasons for concern about the reliability of computer systems.
    3. Computer-related concepts
      - a. Identify basic computer-related terminology.
      - b. Outline the functions of data processing.
      - c. Illustrate the parts of the information processing cycle, and explain how each part is interdependent within the cycle.
      - d. Illustrate parts of the computer processor.
      - e. Compare the similarities and differences among various computer systems.
      - f. Outline the events in history pertinent to breakthroughs in data processing.
      - g. Describe and categorize characteristics of computer languages.
  - B. Computer hardware utilization
    1. Computer hardware
      - a. Identify the types of computers.

- b. Describe primary memory.
  - c. Describe various input devices.
  - d. Identify various output devices.
  - e. Identify other hardware devices.
- 2. Computer software
  - a. Describe the characteristics and functions of an operating system.
  - b. Identify and categorize applications that would utilize batch processing and real-time processing.
  - c. Identify a compiler, interpreter utilities, and virtual storage.
  - d. Discuss the different versions of the microcomputer operating systems.
  - e. Identify the major components of the DOS/VS operating system.
- 3. Computer operation and maintenance
  - a. Identify terminology related to computer operation and maintenance.
  - b. Operate and maintain machines.
  - c. Monitor and report environmental problems.
  - d. Inventory resources and report findings.
  - e. Handle media and support tools.
  - f. Monitor hardware performance.
  - g. Schedule workflow.
  - h. Maintain data and physical security.
- C. Program development
  - 1. Problem-solving concepts and applications
    - a. Define structured programming and related terminology.
    - b. Identify three types of control structures.

- c. Demonstrate problem-solving techniques.
- d. Utilize flowcharting and top-down analysis.
- 2. Programming concepts and applications
  - a. **MODULE A: SETUP PROCEDURES**
    - 1. Demonstrate the ability to power-up, load the compiler, and power-down the computer.
    - 2. List the steps of problem solving.
  - b. **MODULE B: INTRODUCTION TO PROGRAMMING**
    - 1. Identify keywords and commands.
    - 2. Code, key, save, load, and execute a program.
    - 3. Edit program lines.
    - 4. Write simple programs using the keywords REM, PRINT, END.
  - c. **MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING**
    - 1. Explain the benefits of documentation.
    - 2. Identify the three major types of errors.
    - 3. Develop test data for a program.
  - d. **MODULE D: USING ARITHMETIC**
    - 1. Define the types of variables.
    - 2. Use arithmetic operators.
    - 3. Develop arithmetic equations.
    - 4. Write programs using the keyword LET.
  - e. **MODULE E: INPUT CONCEPTS**
    - 1. Demonstrate the use of data terminators.
    - 2. Write programs using the keywords READ/DATA, IF...THEN, GOTO, and INPUT.
    - 3. Demonstrate an output report by using proper headings (major, minor, and column) and PRINT USING.

f. **MODULE F: BRANCHING**

1. Identify the relational operators.
2. Identify the logical operators.
3. Describe the difference between a conditional branch and an unconditional branch.
4. Write programs using the conditional branch.

g. **MODULE G: TOTALS**

1. Define totaling and counting.
2. Demonstrate how subtotals are accumulated and printed.
3. Write programs to demonstrate totaling, counting, and subtotals.

h. **MODULE H: MODULE PROGRAMMING**

1. Define modular programming.
2. Write programs to use a main module and submodules.

i. **MODULE I: CONTROLLED LOOPS**

1. Write programs using the keywords FOR/NEXT.

j. **MODULE J: ADVANCED CONCEPTS**

1. Write programs using tables to store data.
2. Write programs using sort routines.
3. Write programs using advanced character handling.
4. Write programs using graphics.

D. Data storage

1. Media and types of storage

- a. Define primary and secondary storage.
- b. Identify secondary storage devices and media.
- c. Distinguish between volatile and nonvolatile memory.

- d. Differentiate between file organizations and file access methods.
  - 2. Data storage applications
    - a. Identify data elements--field, record, and file.
    - b. Code a program to create a sequential and/or random data file.
    - c. Code a program to retrieve data, and produce a report from a data file they created.
  - 3. Data representation and numbering systems
    - a. Define character and control codes.
    - b. Give examples of ASCII coding.
    - c. Give examples of EBCDIC coding.
    - d. Demonstrate an understanding of the binary numbering system.
    - e. Demonstrate an understanding of the octal numbering system.
    - f. Demonstrate an understanding of the hexadecimal numbering system.
- III. Analysis of student competencies
- IV. Course evaluation
- V. Examination



**ACTIVITIES 1-2: PERSONAL SUCCESS AND THE BUSINESS  
ENVIRONMENT/ANALYSIS OF CONTENT  
IN RELATION TO OBJECTIVES**

**INSTRUCTIONS:** Break into groups designated by your teacher. Your group has been assigned a specific section from Business Computer Programming. After reviewing the units within the section, make an oral report to your class, having each member of the group give a presentation.

For each unit, report:

- A. Section and unit titles.
- B. Major objectives(s).
- C. Synopsis of unit.
- D. Importance of the unit relative to personal success and the office environment.

### ACTIVITY 3: RECOGNIZING LEVELS OF PERFORMANCE

**INSTRUCTIONS:** From the list of the Business Computer Programming course competencies, rate your personal achievement for each of the competencies.

Circle a 4 if you feel that you have **mastered** the competency.

Circle a 3 if you feel that you have an **average mastery** of the competency.

Circle a 2 if you feel that you have a **fair mastery** of the competency.

Circle a 1 if you feel that you have only a **minimal mastery** or if you have not had the opportunity to master the competency.

## COMPETENCY PERFORMANCE EVALUATION

(Maximum = 4 and Minimum = 1)

### COMPETENCIES

#### I. Introduction to Business Computer Programming

##### 1. Introduction To Course

- 4 3 2 1 1. Relate the purpose of Business Computer Programming to office education, business, and industry.
- 4 3 2 1 2. Recognize the role of the youth organization relative to Business Computer Programming.
- 4 3 2 1 3. Discuss course objectives, equipment, units, competencies, and evaluation measures.
- 4 3 2 1 4. Explain classroom organization, management, safety, and evaluation measures.
- 4 3 2 1 5. Set realistic long- and short-range goals.
- 4 3 2 1 6. Update study skills needed to reach realistic goals.

##### 2. Social Implications

- 4 3 2 1 1. Identify terminology related to computer crime, computer users, and new computer devices and techniques.
- 4 3 2 1 2. Identify the major issues and technological developments related to computers and society.
- 4 3 2 1 3. Describe the ethical and legal issues that have arisen as a result of the computer revolution.
- 4 3 2 1 4. Identify privacy issues involved with electronic data storage.
- 4 3 2 1 5. Explain the reasons for concern about the reliability of computer systems.

##### 3. Computer-Related Concepts

- 4 3 2 1 1. Identify basic computer-related terminology.
- 4 3 2 1 2. Outline the functions of data processing.
- 4 3 2 1 3. Illustrate the parts of the information processing cycle, and explain how each part is interdependent within the cycle.
- 4 3 2 1 4. Illustrate parts of the computer processor.
- 4 3 2 1 5. Compare the similarities and differences among various computer systems.
- 4 3 2 1 6. Outline the events in history pertinent to breakthroughs in data processing.
- 4 3 2 1 7. Describe and categorize characteristics of computer languages.

#### II. Computer Hardware Utilization

##### 1. Computer Hardware

- 4 3 2 1 1. Identify the types of computers.
- 4 3 2 1 2. Describe primary memory.
- 4 3 2 1 3. Describe various input devices.
- 4 3 2 1 4. Identify various output devices.
- 4 3 2 1 5. Identify other hardware devices.

### COMPETENCIES

#### 2. Computer Software

- 4 3 2 1 1. Describe the characteristics and functions of an operating system.
- 4 3 2 1 2. Identify and categorize applications that would utilize batch processing and real-time processing.
- 4 3 2 1 3. Identify a compiler, interpreter utilities, and virtual storage.
- 4 3 2 1 4. Discuss the different versions of the microcomputer operating systems.
- 4 3 2 1 5. Identify the major components of the DOS/VS operating system.

#### 3. Computer Operation and Maintenance

- 4 3 2 1 1. Identify terminology related to computer operation and maintenance.
- 4 3 2 1 2. Operate and maintain machines.
- 4 3 2 1 3. Monitor and report environmental problems.
- 4 3 2 1 4. Inventory resources and report findings.
- 4 3 2 1 5. Handle media and support tools.
- 4 3 2 1 6. Monitor hardware performance.
- 4 3 2 1 7. Schedule workflow.
- 4 3 2 1 8. Maintain data and physical security.

### III. Program Development

#### 1. Problem-Solving Concepts and Applications

- 4 3 2 1 1. Define structured programming and related terminology.
- 4 3 2 1 2. Identify three types of control structures.
- 4 3 2 1 3. Demonstrate problem-solving techniques.
- 4 3 2 1 4. Utilize flowcharting and top-down analysis.

#### 2. Programming Concepts and Applications

##### MODULE A: SETUP PROCEDURES

- 4 3 2 1 1. Demonstrate the ability to power-up, load the compiler, and power-down the computer.
- 4 3 2 1 2. List the steps of problem solving.

##### MODULE B: INTRODUCTION TO PROGRAMMING

- 4 3 2 1 1. Identify keywords and commands.
- 4 3 2 1 2. Code, key, save, load, and execute a program.
- 4 3 2 1 3. Edit program lines.
- 4 3 2 1 4. Write simple programs using the keywords REM, PRINT, END.

##### MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING

- 4 3 2 1 1. Explain the benefits of documentation.
- 4 3 2 1 2. Identify the three major types of errors.
- 4 3 2 1 3. Develop test data for a program.

### COMPETENCIES

#### *MODULE D: USING ARITHMETIC*

- 4 3 2 1 1. Define the types of variables.
- 4 3 2 1 2. Use arithmetic operators.
- 4 3 2 1 3. Develop arithmetic equations.
- 4 3 2 1 4. Write programs using the keyword LET.

#### *MODULE E: INPUT CONCEPTS*

- 4 3 2 1 1. Demonstrate the use of data terminators.
- 4 3 2 1 2. Write programs using the keywords READ/DATA, IF...THEN, GOTO, and INPUT.
- 4 3 2 1 3. Demonstrate an output report by using proper headings (major, minor, and column) and PRINT USING.

#### *MODULE F: BRANCHING*

- 4 3 2 1 1. Identify the relational operators.
- 4 3 2 1 2. Identify the logical operators.
- 4 3 2 1 3. Describe the difference between a conditional branch and an unconditional branch.
- 4 3 2 1 4. Write programs using the conditional branch.

#### *MODULE G: TOTALS*

- 4 3 2 1 1. Define totaling and counting.
- 4 3 2 1 2. Demonstrate how subtotals are accumulated and printed.
- 4 3 2 1 3. Write programs to demonstrate totaling, counting, and subtotals.

#### *MODULE H: MODULE PROGRAMMING*

- 4 3 2 1 1. Define modular programming.
- 4 3 2 1 2. Write programs to use a main module and submodules.

#### *MODULE I: CONTROLLED LOOPS*

- 4 3 2 1 1. Write programs using the keywords FOR/NEXT.

### COMPETENCIES

#### *MODULE J: ADVANCED CONCEPTS*

- 4 3 2 1 1. Write programs using tables to store data.
- 4 3 2 1 2. Write programs using sort routines.
- 4 3 2 1 3. Write programs using advanced character handling.
- 4 3 2 1 4. Write programs using graphics.

### *IV. Data Storage*

#### *1. Media and Types of Storage*

- 4 3 2 1 1. Define primary and secondary storage.
- 4 3 2 1 2. Identify secondary storage devices and media.
- 4 3 2 1 3. Distinguish between volatile and nonvolatile memory.
- 4 3 2 1 4. Differentiate between file organizations and file access methods.

#### *2. Data Storage Applications*

- 4 3 2 1 1. Identify data elements--field, record, and file.
- 4 3 2 1 2. Code a program to create a sequential and/or random data file.
- 4 3 2 1 3. Code a program to retrieve data, and produce a report from a data file they created.

#### *3. Data Representation and Numbering Systems*

- 4 3 2 1 1. Define character and control codes.
- 4 3 2 1 2. Give examples of ASCII coding.
- 4 3 2 1 3. Give examples of EBCDIC coding.
- 4 3 2 1 4. Demonstrate an understanding of the binary numbering system.
- 4 3 2 1 5. Demonstrate an understanding of the octal numbering system.
- 4 3 2 1 6. Demonstrate an understanding of the hexadecimal numbering system.

#### ACTIVITY 4: ASSESSING THE OBJECTIVES AND COURSE CONTENT

**INSTRUCTIONS:** Working in the same group and section and units as in Activity 1, assign each member of the group a specific portion of a section. Have each group member prepare seven review questions with answers which could be used as examination questions. As part of the seven questions, include two that relate to skill development. Key in the questions following the instructions provided by your teacher so that they can be printed out and copied for your classmates.

## REFERENCE ADDRESSES

### LIST OF PUBLISHERS

ADDISON-WESLEY PUBLISHING  
CO., INC.  
Reading, MA 01867

ALLYN & BACON  
College Division  
Rockleigh, NJ 07647

AMERICAN BANKERS ASSN.  
1120 Connecticut Ave. NW  
Washington, DC 20036

ANAHEIM PUBLISHING CO.  
2632 Saturn Street  
Brea, CA 92621

ARCO PUBLISHING, INC.  
Educational Division  
215 Park Avenue South  
New York, NY 10003

AVA RESOURCES  
2020 N. 14th Street  
Arlington, VA 22201

BANTAM BOOKS  
666 Fifth Avenue  
New York, NY 10103

BOBBS-MERRILL EDUCATIONAL  
PUBLISHERS  
P.O. Box 7080  
Indianapolis, IN 46206

ROBERT J. BRADY, CO.  
A Prentice-Hall Co.  
Bowie, MD 20715

BROOKS/COLE PUBLISHING CO.  
540 Arago Street  
Monterey, CA 93940

WILLIAM C. BROWN PUBLISHING  
2460 Kerper Blvd.  
Dubuque, IA 52001

CAREER AIDS, INC.  
0417 Nordhoff St.  
Dept. K 65  
Chatsworth, CA 91311

CBS COLLEGE PUBLISHING  
383 Madison Avenue  
New York, NY 10017

CHILTON BOOK COMPANY  
School and Library Dept  
Radnor, PA 19089

CHRONICLE GUIDANCE PUBL., INC  
Aurora Street Extension  
P.O. Box 1190  
Moravia, NY 13118-1190

COLLEGE BOARD PUBLICATIONS  
The College Board  
888 Seventh Ave.  
New York, NY 10106

CURRICULUM PUBLICATIONS  
CLEARINGHOUSE  
46 Horrbin Hall  
Macomb, IL 61455

CURTIN & LONDON, INC.  
135 West 50th Street  
New York, NY 10020

DATAPRO  
1805 Underwood Blvd.  
Delran, NJ 08075

DELMAR PUBLISHING INC.  
2 Computer Drive West  
Box 15-015  
Albany, NY 12212

THE DRYDEN PRESS  
P.O. Box 36  
Lavallette, NJ 08735

EDITS  
P.O. Box 7234  
San Diego, CA 92107

EDUCATIONAL FILMSTRIPS  
1405 19th Street  
Huntsville, TX 77340

GLENCOE PUBLISHING CO.  
8301 Ambassador Row  
Dallas, TX 75247

**Business Computer Programming I**  
**APPENDIX A: Reference Addresses**

GLENCOE PUBLISHING CO.  
17337 Ventura Blvd.  
Encino, CA 91316

GREGG/MCGRAW HILL  
13955 Manchester Road  
Manchester, MD 63011

GREGG DIVISION  
McGraw-Hill Book Co.  
1221 Avenue of the Americas  
New York, NY 10020

HARPER & ROW  
6780 Sierra Court  
Suite N  
Dublin, CA 94566

HARPER & ROW  
Keystone Ind. Park  
Scranton, PA 18512

HBJ/MEDIA SYSTEMS CORP  
757 Third Avenue  
New York, NY 10164

HEMPHILL  
1400 Wather  
Austin, TX 78703

HOUGHTON-MIFFLIN CO.  
One Beacon Street  
Boston, MA 02107

IRWIN PUBLISHING CO  
1818 Ridge Road  
Homewood, IL 60430

MPC EDUCATIONAL PUBLISHERS  
839 White Plains Road  
Bronx, NY 10467

MAYFIELD PUBLISHING CO.  
285 Hamilton Avenue  
Palo Alto, CA 94301

MEDIA SYSTEMS, INC.  
757 Third Avenue  
New York, NY 10017

CHARLES E. MERRILL PUBL. CO.  
1200 Alum Creek Drive  
Columbus, OH 43216

MILLIKEN PUBLISHING CO.  
Box 21579  
St. Louis, MO 63132

NATIONAL OCCUPATIONAL CO.  
TESTING INSTITUTE  
318 Johnson Hall  
Ferris State College  
Big Rapids, MI

NATIONAL PUBLIC RADIO  
Cassette Catalog  
Cassette Publishing  
2025 M Street N.W.  
Washington, DC 20036

NATIONAL TEXTBOOK CO.  
4255 West Touhy Avenue  
Lincolnwood, IL 60066-1975

OHIO STATE UNIVERSITY  
Vocational Instructional  
Materials Laboratory  
154 W. Twelfth Ave.  
Columbus, OH 43210-1302

PEARSON  
9219 Chimney Corner  
Dallas, TX 75243

PRENTICE-HALL, INC.  
Englewood Cliffs, NJ 07632

PRENTICE-HALL MEDIA  
ServCode DX  
150 White Plains Rd.  
Tarrytown, NY 10591

RADIO SHACK EDUCATION DIVISION  
1400 One Tandy Center  
Ft. Worth, TX 76102-2805

RANDOM HOUSE SCHOOL DIVISION  
Dept. 9232  
400 Hahn Road  
Westminister, MD 21157

RESTON PUBLISHING CO.  
A Prentice-Hall Co.  
11480 Sunset Hills Road  
Reston, VA 22090

THE H. M. ROWE CO.  
624 N. Gilmore Street  
Baltimore, MD 21217

**Business Computer Programming I**  
**APPENDIX A: Reference Addresses**

SCIENCE RESEARCH ASSN., INC.  
College & Vocational Studies  
155 N. Wacker Drive  
Chicago, IL 60606

SOUTHWESTERN PUBLISHING CO.  
5101 Madison Road  
Cincinnati, OH 45227

STATEWIDE TEACHING AIDS  
5845 Gardner Hwy.  
Alexandria, LA 71306

TAB BOOKS  
Blue Ridge  
Summit, PA 17214

TAR EDUCATION  
P.O. Box 14488  
Austin, TX 78761

TEACHING AIDS INC.  
P.O. Box 1798  
Costa Mesa, CA 92628-0798

VOTECH-OKLAHOMA STATE DEPT.  
OF VOCATIONAL AND TECHNICAL  
EDUCATION  
1515 West Sixth Avenue  
Stillwater, OK 74074

V TECS - COMMISSION ON  
EDUCATION INSTITUTIONS  
Southern Assn. Of Colleges and Schools  
795 Peachtree Street  
N.E. Atlanta, GA 30365

WADSWORTH  
7625 Empire Drive  
Florence, KY 41402

WEST PUBLISHING  
50 Kellogg Blvd.  
St. Paul, MN 55102

WESTERN TAPE  
2761 Marine Way  
Mountain View, CA 94040

J. WESTON WALCH, PUBL.  
P.O. Box 658  
Portland, MA 04104-0658

JOHN WILEY AND SONS, INC.  
605 Third Avenue  
New York, NY 10158

**SOFTWARE VENDORS**

ABBOT EDUCATIONAL SOFTWARE  
334 Westwood Avenue  
East Longmeadow, MA 01028  
(413)525-3462

ACADEMIC SOFTWARE  
22 E. Quackenbush Avenue  
Durmont, NJ 07628

ACORN SOFTWARE PRODUCTS  
34 North Carolina Avenue, S.E.  
Washington, D.C. 20003  
(202)544-4259

ACTIVISION, INC.  
P.O. Box 7287  
Mountain View, CA 94030

ADDISON-WESLEY  
2725 Sand Hill Road  
Menlo Park, CA 94025

ADDISON-WESLEY PUBLISHING CO.  
Business & Professional Division  
Reading, MA 01867  
(617)944-3700

ADVANCED SYSTEMS, INC.  
2340 S. Arlington Heights Rd.  
Arlington Heights, IL 60005  
(312)981-4260

ALDUS  
411 First Ave., S.  
Suite 200  
Seattle, WA 98136

ALTERNATE SOURCE  
1806 Ada Street  
Lansing, MI 48910  
(517)487-3358

AMERICAN TRAINING  
INTERNATIONAL  
3770 Highland Ave.  
Suite 201 B  
Manhattan Beach, CA 90266  
(213)546-4725



Business Computer Programming I  
APPENDIX A: Reference Addresses

APPLE COMPUTER COMPANY  
10260 Bandley Drive  
Cupertino, CA 95014  
(408)996-1010

APPLICATIONS  
10260 W. Eleven Mile Road  
Suite 103  
Southfield, MI 48076

APPLIED EDUCATIONAL SYSTEMS  
RFD #2, Box 213  
Dunbarton, NH 03301  
(603)774-6151

APPLIED ENGINEERING  
P.O. Box 798  
Carrollton, TX 75006

APPLIED MICROSYSTEMS  
P.O. Box 832  
Roswell, GA 30077

APPLIED SOFTWARE TECHNOLOGY  
14125 Capli Drive  
Los Gatos, CA 95030

AURORA SYSTEMS  
2040 East Washington Avenue  
Madison, WI 53704  
(608)249-5875

B & B SOFTWARE  
P.O. Box 2090  
Ann Arbor, MI 48106  
(313)668-8152

BELL & HOWELL MICRO SYSTEMS  
Audio Visual Products Division  
7100 North McCormick Road  
Chicago, IL 60645  
(312)262-1600

BETAMAX, INC.  
101 Nickerson Street  
Suite 550  
Seattle, WA 98109  
(206)282-6249

BITS & BYTES  
Fenner Block, 33 Church Street  
Fredonia, NY 14063  
(716)673-1515

BORG-WARNER EDUCATIONAL  
SYSTEMS  
600 West University Drive  
Arlington Heights, IL 60004  
(800)323-7577

BRINKER COMPUTING  
2775 Tessmer Rd.  
Ann Arbor, MI 48103  
(313)662-6386

BRODERBUND  
17 Paul Drive  
San Rafael, CA 94903-2101

BUSINESS AND AGRI SERVICES  
R.R. 2, Box 118  
Raymond, IL 62560  
(217)229-3584

CADKEY INC.  
27 Hartford Turnpike  
Vernon, CT 06066

CAREER AIDS  
20417 Nordhoff St., Dept. F5  
Chatsworth, CA 91311

CAREER PUBLISHING INCORPORATED  
P.O. Box 5486  
Orange, CA 92613-5486

COMPREHENSIVE SOFTWARE  
2316 Artesia Blvd  
Suite B  
Redondo Beach, CA 90278  
(213)318-2561

COMPUSERVE  
5000 Arlington Center Blvd.  
Columbus, OH 43220

COMPUTATION  
P.O. Box 502  
Troy, MI 48099

COMPU-TEACH  
240 Bradley St.  
New Haven, CT 06511

THE COMPUTER CENTER  
302 Commercial  
Waterloo, IA 50701  
(319)232-9504

Business Computer Programming I  
APPENDIX A: Reference Addresses

COMPUTER INFORMATION  
EXCHANGE

P.O. Box 159  
San Luis Rey, CA 92068  
(714)757-4849

COMPUTERLAND  
2500-B South Columbia Road  
Grand Forks, ND 55082  
(612)436-7198

COMPUTERS PLUS COMPANY  
2303 N. 44th St. #2  
Phoenix, AZ 85008  
(602)955-1404

COMPUTREX COMPUTER SERVICES  
P.O. Box 536  
Inman, SC 29349

COMPUTER HOUSE DIVISION  
1407 Clinton Road  
Jackson, MI 49202  
(517)782-2132

CONTINENTAL SOFTWARE  
12101 Jefferson Blvd.  
Culver City, CA 90230  
(213)371-5612

CONTROL DATA CORPORATION  
P.O. Box 261127  
San Diego, CA 92126  
(800)233-3784

COOK'S COMPUTER COMPANY  
1905 Bailey Drive  
Marshalltown, IA 50158

COURSEWARE, INC.  
10075 Carroll Canyon Rd.  
San Diego, CA 92131  
(619)578-1700

CREATING COMPUTING SOFTWARE  
Department 301  
P.O. Box 789M  
Morristown, NJ 07960

CREATIVE SOFTWARE  
230 E. Caribbean Dr.  
Sunnyvale, CA 94089

CURRICULUM APPLICATIONS  
P.O. BOX 264  
Arlington, MA 02174

DALLAS PUBLIC SCHOOLS  
Marketing Department  
912 South Ervay  
Dallas, TX 75201  
(214)742-7991

DECISION DATA, INC.  
213 Lincoln Way  
Ames, IA 50010  
(515)233-4807

DESIGN SOFTWARE INC.  
Learning Services  
61 West Eighth St.  
P.O. Box 10636  
Eugene, OR 97440-2636

DESIGNWARE  
345 Fourth St.  
San Francisco, CA 94107

DISKCOUNT DATA  
2701-C W. 15th  
Suite 612  
Plano, TX 75075

DORSETTE EDUCATIONAL  
SYSTEMS, INC.  
P.O. Box 1226  
Norman, OK 73070  
(405)288-2300

DRW EDUCATIONAL SYSTEMS  
P.O. Box 2941  
Costa Mesa, CA 92628  
(714)548-9754

EDUCATIONAL ACTIVITIES, INC.  
P.O. Box 392  
Freeport, NY 11520  
(800)645-3739

EDUCATIONAL COURSEWARE  
Ten Bay Street  
Design 66  
Westport, CT 06880

EDUCATIONAL MICROSYSTEMS, INC.  
P.O. Box 471  
Chester, NJ 07930  
(210)879-5982

Business Computer Programming I  
APPENDIX A: Reference Addresses

EDUCATIONAL PROGRAMMING  
SYSTEMS

1328 Baus Boulevard  
St. Louis, MO 63132  
(314)991-0300

EDUCATIONAL PROGRAMS

P.O. Box 2345  
West Lafayette, IN 47906  
(317)463-4778

EDUCATIONAL RESOURCES

2360 Hassell Road  
Hoffman Estates, IL 60195

EDUCATIONAL SERVICES  
MANAGEMENT

P.O. Box 12599  
Research Triangle Park, NC 27709  
(919)781-1500

EDUCATIONAL SOFTWARE CORP.

414 Rosemere  
Maquoketa, IA 52060

EDUCATIONAL SOFTWARE  
AND DESIGN

P.O. Box 2801  
Flagstaff, AZ 86003

EDUTECH

50 Putnam Street  
West Newton, MA 02165  
(617)965-4813

EDUTEK CORP.

P.O. Box 11354  
Palo Alto, CA 94306  
(415)325-9965

EDUCATIONAL TESTING SERVICES

SIGI Office  
Princeton, NJ 08541  
(609)734-5165

EDU-WARE

P.O. Box 336  
Maynard, MA 01754

EDU-WARE SERVICES, INC.

22222 Sherman Way  
Suite 102  
Canoga Park, CA 91303  
(213)346-6783

ELECTRONIC ARTS

1820 Gateway Drive  
San Mateo, CA 94404

EMC PUBLISHING

Changing Times Education Services  
300 York Avenue  
St. Paul, MN 55101

EPYX

600 Galveston Drive  
Redwood, CA

GENERIC SOFTWARE

8763 148th NE  
Redmond, WA 98052

GREAT PLAINS COMPUTERS

113 Broadway  
Fargo, ND 58102  
(701)293-8483

HARRIS LABORATORIES

624 Peach St., P.O. Box 8037  
Lincoln, NE 68501  
(402)476-2811

HARTLEY COURSEWARE, INC.

P.O. Box 431  
Diamondale, MI 48821  
(616)942-8987

HARVEST COMPUTER SYSTEMS

203 West 11th Street  
Alexandria, IN 46001  
(317)724-9527

HAYDEN BOOK COMPANY

50 Essex Street  
Rochelle Park, NJ 07662  
(800)631-0856

HEFFERNAN ONE-SOURCE

2105 Donley Drive  
Austin, TX 78753

HIGH TECHNOLOGY SOFTWARE  
PRODUCTS, INC.

P.O. Box 14665  
Oklahoma City, OK 73113  
(405)840-9900

HORIZONS UNLIMITED  
5367 Sage  
Edwards, CA 93523  
(805)258-8412

HOUGHTON-MIFFLIN  
Educational Software Division  
P.O. Box 683  
Hanover, NH 03755

INDIVIDUAL SOFTWARE, INC.  
1163-1 Chess Dr.  
Foster City, CA 94404  
(415)341-6116

INNOVATIVE PROGRAMMING  
ASSOCIATES  
One Airport Place  
Princeton, NJ 08540  
(609)924-7272

INSTRUCTIONAL SOFTWARE, INC.  
1278 Massachusetts Ave.  
Cambridge, MA 02138  
(617)492-0386

INSTRUCTOR'S COMPUTER  
DIRECTORY  
Instructor Publications  
545 5th Ave.  
New York, NY 10017

INSTRUCTOR'S COMPUTER  
DIRECTORY FOR SCHOOLS  
P.O. Box 6338  
Duluth, MN 55806-9946  
(212)503-2888

INTERNATIONAL BUREAU OF  
SOFTWARE TESTING  
536 Wendell, Suite 7  
Sunnyvale, CA 94086

INTERNATIONAL MICRO SYSTEMS  
8425 Quivara Road  
Lenexa, KS 66215  
(913)888-8330

INTERPRETIVE EDUCATION  
2306 Winters Drive  
Kalamazoo, MI 49002  
(616)345-8681

INTERPRETIVE EDUCATION  
INCORPORATED  
157 S. Kalamazoo Mall  
Suite 250  
Kalamazoo, MI 49007  
(616)345-8681

IQUEST  
CompuServe  
5000 Arlington Center Blvd.  
Columbus, OH 43220

KNOWARE, INC.  
301 Vassar Street  
Cambridge, MA 02139  
(617)576-3821

KRELL SOFTWARE  
21 Millbrook Drive  
Stony Brook, NY 11790  
(518)751-5139

LEARNING ARTS  
P.O. Box 179  
Wichita, KS 67201  
(316)682-6594

LEARNING SERVICES  
61 West Eighth St.  
P.O. Box 10636  
Eugene, OR 97440-2636

LIBRARY SOFTWARE  
P.O. Box 23897  
Pleasant Hill, CA 94523

LOTUS DEVELOPMENT CORP.  
55 Cambridge Parkway  
Cambridge, MA 02142

CHARLES MANN & ASSOCIATES  
Microcomputer Division  
55722 Santa Fe Trail  
Yucca Valley, CA 92284  
(714)365-9718

MCGRAW-HILL BOOK COMPANY  
School Division  
1221 Avenue of the Americas  
New York, NY 10020  
(800)223-4180

Business Computer Programming I  
APPENDIX A: Reference Addresses

MECC

3490 Lexington Ave. North  
Saint Paul, MN 55126-8097

MICRO LAB LEARNING CENTER

3218 Skokie Valley Road  
Highland Park, IL 60035  
(312)433-7550

MICRO LEARNINGWARE

P.O. Box 2134  
North Mankato, MN 56001  
(507)625-2205

MICRO-COMP INCORPORATED

2015 N.W. Circle Blvd.  
Corvallis, OR 97330

MICROCOMPUTER EDUCATION

Applications Network  
256 N. Washington Street  
Falls Church, VA 22046  
(702)536-2310

MICROCOMPUTERS IN EDUCATION

4148 Winnetka Avenue, North  
Minneapolis, MN 55427  
(612)533-2781

MICROCOMPUTER SOFTWARE

Illinois Vocational Curriculum Center  
Sangamon State University, E-22  
Springfield, IL 62708  
(217)786-6375

MICROCOMPUTER SOFTWARE  
SYSTEMS

4716 Lakewood Drive  
Metairie, LA 70002

MICRO-ED, INC.

P.O. Box 24156  
Minneapolis, MN 55424  
(612)926-2292

MICROPUTE SOFTWARE

P.O. Box 1943  
Rock Mount, NC 27801

MICROSOFT

16011 NE 36th Way  
P.O. Box 97017  
Redmond, WA 98052

MICROSOFT CONSUMER PRODUCTS

400 180th Avenue, N.E.  
Bellevue, WA 98004  
(206)454-1313

MILLIKEN PUBLISHING COMPANY

1100 Research Boulevard  
St. Louis, MO 63132  
(314)991-4220

MINNESOTA CURRICULUM SERVICES  
CENTER

3554 White Bear Avenue  
White Bear Lake, MN 55110  
(612)770-3943

MINDSCAPE

3444 Dundee Road  
Northbrook, IL 60015

MINNESOTA EDUCATIONAL  
COMPUTING

Consortium Publications  
2520 Broadway Drive  
St. Paul, MN 55113  
(800)631-8112

MUSE SOFTWARE

347 N. Charles St.  
Baltimore, MD 21201  
(301)659-7212

NCCD (National Coordinating Center  
for Curriculum Development)

State University of New York  
Stony Brook, NY 11794  
(516)246-8418

NATIONAL SOFTWARE MARKETING

4701 McKinley Street  
Hollywood, FL 33021  
(305)625-6062

NORTON COMPUTING

2210 Wilshire Blvd. #186  
Santa Monica, CA 90403

NUTMEG SYSTEMS, INC.

25 South Ave.  
New Canaan, CT 06840

ON-LINE SYSTEMS

36575 Mudge Ranch Road  
Coarsegold, CA 93614  
(202)683-6858

Business Computer Programming I  
APPENDIX A: Reference Addresses

OPTICAL DATA CORP.  
66 Hanover Road  
Box 97  
Florham Park, NJ 07932

PHOTOCOM PRODUCTIONS  
P.O. Box 3135  
Pismo Beach, CA 93449  
(800)321-7367

PILLSBURY CENTER  
Mail Center 3286  
200 S. 6th Street  
Minneapolis, MN 55402  
(612)330-8732

PRECISION PEOPLE, INC.  
P.O. Box 17402  
Jacksonville, FL 32216  
(904)642-1980

PROGRAMS FOR LEARNING, INC.  
P.O. Box 954  
New Millford, CT 06776

QUALITY EDUCATIONAL DESIGNS  
P.O. Box 12486  
Portland, OR 97212  
(503)282-4906

RAINBOW MICRO SOFTWARE  
1650 Piikea Street  
Honolulu, HI 96818  
(TRS-80 Programs)

RANDOM HOUSE MEDIA GROUP  
201 E. 50th St.  
New York, NY 10017

REGENCY EDUCATIONAL SYSTEMS  
4951 Airport Parkway  
Dallas, TX 75248

RESTON PUBLISHING  
11480 Sunset Hills Rd.  
Reston, VA 22190  
(703)437-8900

SCHOLASTIC, INC.  
904 Sylvan Ave.  
Englewood Cliffs, NJ 07632

SCHOOL MICROWARE  
P.O. Box 246  
Dresden, ME 04342  
(207)737-4466

SILICON EXPRESS  
955 E. Main Street  
Columbus, OH 43213

SOCIAL SCIENCE SOFTWARE  
17408 N.E. 19th Place  
Bellevue, WA 98008  
(206)747-0948

SOFTWARE EXCHANGE  
6 South Street, Box 68  
Milford, NH 03055  
(603)673-5144

SOFTWARE INDUSTRIES  
902 Pinecrest  
Richardson, TX 75080

SOFTWARE PUBLISHING CORP.  
1901 Landings Drive  
Mountain View, CA 94043

SOUTHERN COMPUTER SYSTEMS, INC.  
P.O. Box 55473  
Birmingham, AL 35255

SOUTH-WESTERN PUBLISHING  
COMPANY  
5101 Madison Rd.  
Cincinnati, OH 45227  
(513)271-8811

SPRINGBOARD SOFTWARE, INC.  
7808 CreekrIDGE Circle  
Minneapolis, MN 55435

JOHN STEFFEE, INC.  
1030 Buffalo Trail  
Tyler, TX 75703

STERLING SWIFT PUBLISHING  
COMPANY  
7901 South IH-35  
Austin, TX 78744  
(512)282-6840



STONEWARE

Microcomputer Products  
50 Belvedere Street  
San Rafael, CA 94901  
(415)454-6500

STRICTLY SOFTWARE

4321 N. 39th Street  
Phoenix, AZ 85018  
(602)955-8138

SUNBURST COMMUNICATIONS

39 Washington Avenue  
Pleasantville, NY 10570

TEACHER'S PET

c/o Glenn Fisher  
1517 Holly Street  
Berkeley, CA 94703

THE TEACHING ASSISTANT

22 Seward Drive  
Huntington Station, NY 11746

TEACHING TOOLS

P.O. Box 12679  
Research Triangle Park, NC 27709  
(919)851-2374

TIME SHARE CORP

Hanover, NH 03775  
(603)488-3838

TIMEWORKS, INC.

444 Lake Cook Rd.  
Deerfield, IL 60015

TYC SOFTWARE

40 Stuyvesant Manor  
Geneseo, NY 14454  
(716)243-3005

VERSA CAD CORP.

7372 Prince Drive  
Huntington Beach, CA 92647

VISICORP

2895 Zanker Rd.  
San Jose, CA 95134  
(408)946-9000

VOCATIONAL MEDIA ASSOCIATES  
Mount Kisco, NY 10549

WALT DISNEY EDUCATIONAL  
MEDIA CO.

500 S. Buena Vista St.  
Burbank, CA 91521

WESTINGHOUSE LEARNING CORP.

5505 West 110th Street  
Oak Lawn, IL 60453

J. WESTON WALCH, PUBLISHER

321 Valley St., P.O. Box 658  
Portland, ME 04104-0658

WHITNEY EDUCATIONAL SERVICES

2071 Tenth Avenue  
San Francisco, CA 94116

WORDPERFECT

1555 North Technology Way  
Orem, UT 84057

MICROCOMPUTER DIRECTORIES

*APPLE COMPUTER EDUCATION  
SOFTWARE DIRECTORY*

Visual Materials, Inc.

4170 Grove Avenue  
Gurnee, IL 60031

Describes nearly 1,100 software packages in  
19 subject categories from elementary  
through college level. Also includes ad-  
ministrative, office, and utility applications  
packages.

*APPLE II/III SOFTWARE DIRECTORY*

Vital Information, Inc.

7899 Mastin Drive  
Overland Park, KS 66204

Program descriptions of Apple software for  
home, business, and education. Also includes  
a vendor list and list of Apple user groups.

*THE APPLE SOFTWARE DIRECTORY*

WIDL Video

5245 W. Diversey Ave.  
Chicago, IL 60639

(312)622-9606

Educational software from 400 vendors by  
subject matter.

*THE BOOK OF APPLE SOFTWARE*

The Book Company  
11223 So. Hindry Ave.  
Los Angeles, CA 90045  
(213)417-8031

**or available from:**

Addison-Wesley Publishing Company  
Reading, MA 01867.

Approximately 125 educational programs reviewed by paragraph descriptions and numerical ratings. Field test results are also available.

*THE COMMODORE SOFTWARE  
ENCYCLOPEDIA*

Commodore Corporate Offices  
Educational Department  
487 Devon Park Drive  
Wayne, PA 19087  
(215)687-9750

Available at local Commodore dealers or from software publishers. Special education section lists approximately 740 programs from various software publishers. Only Commodore-approved software listed.

*EDUCATIONAL SOFTWARE DIRECTORY*

Sterling Swift Publishing Co.  
1600 Fortview Road  
Austin, TX 78704  
Apple software only.

*EDUCATORS HANDBOOK AND  
SOFTWARE DIRECTORY*

Vital Information, Inc.  
7899 Mastin Drive  
Overland Park, KS 66024  
Contains 1,500 program descriptions in 12 major subject categories indexed by title, subject, grade level, and publisher. Also includes articles, a guide to obtaining funding and grants, a list of user groups, and more.

*GAMCO MICROCOMPUTER SOFTWARE*

Box 310 P  
Big Spring, TX 79720-0120  
(915)267-6327  
Contains lists and descriptions by microcomputer and subject area. For Apple, PET, TRS-80

*INTERNATIONAL MICROCOMPUTER  
SOFTWARE DIRECTORY*

Imprint Software  
420 So. Howest St.  
Ft. Collins, CO 80521  
(800)525-5992  
Lists, cross-references over 5,000 programs.

*PC WORLD*

555 DeHaro Street  
San Francisco, CA 94107  
Comprehensive list of products available for the IBM-PC or IBM-PC compatible computers.

*REFERENCE MANUAL FOR INSTRUCTIONAL  
USE OF MICROCOMPUTERS*

Jem Research  
Discovery Park  
University of Victoria  
P.O. Box 100  
Victoria, BC V8W 2Y2, Canada  
(604)477-7246  
Cross references over 1,000 educational software programs; Some programs are evaluated.

*THE SOFTWARE DIRECTORY*

Software Central  
P.O. Box 30424  
Lincoln, NE 68503  
Includes a special section of educational software; Programs are briefly described.

**ELECTRONIC INFORMATION  
SERVICES**

BRS/SEARCH  
1200 Route 7  
Latham, NY 12110  
(800)227-5277

**COMPUSERVE INFORMATION  
SERVICE, INC.**

5000 Arlington Center Blvd.  
Columbus, OH 43220  
(800)848-8990

DATEXT, INC.  
444 Washington St.  
Woburn, MA 01801



DIALOG INFORMATION SERVICE, INC.  
3460 Hillview Avenue  
Palo Alto, CA 94304  
(800)334-2564

DOW JONES/RETRIEVAL SERVICE  
P.O. Box 300  
Princeton, NJ 08540  
(609)452-2000

EASYPNET  
Telebase Systems, Inc.  
134 North Narberth Avenue  
Narberth, PA 19072

GENERAL VIDEOTEXT CORP. (DELPHI)  
3 Blackstone Street  
Cambridge, MA 02139

INFOTRAC  
Information Access Company  
11 Davis Drive  
Belmont, CA 94002

NEWSNET  
945 Haverford Rd.  
Bryn Mawr, PA 19010  
(800)345-1301

NEXIS/LEXIS  
Mead Data Central  
9393 Springboro Pike  
P.O. Box 933  
Dayton, OH 45401  
(800)227-4908

ORBIT Search Service  
SDC Information Service  
2500 Colorado Avenue  
Santa Monica, CA 90406

THE SOURCE  
1616 Anderson Rd.  
McLean, VA 22102  
(800)336-3330

VUTEXT INFORMATION SERVICES,  
INC.  
1211 Chestnut Street  
Philadelphia, PA 19107  
(215)665-3300

## MAGAZINES/PERIODICALS/ NEWSLETTERS

*ADMINISTRATIVE MANAGEMENT*  
1123 Broadway  
New York, NY 10010

*APPLE ORCHARD*  
908 George Street  
Santa Clara, CA 95050

*BUSINESS COMPUTER SYSTEMS*  
Cahners Publishing Company, Inc.  
221 Columbus Avenue  
Boston, MA 02116

*BUSINESS SOFTWARE*  
M & T Publishing  
2464 Embarcadero Way  
Palo Alto, CA 94303

*BUSINESS WEEK*  
McGraw-Hill Publications Company  
1221 Avenue of the Americas  
New York, NY 10020

*BYTE: THE SMALL SYSTEMS JOURNAL*  
70 Main Street  
Peterborough, NH 03458

*CLASSROOM COMPUTER LEARNING*  
Peter Li, Inc.  
2451 East River Rd.  
Dayton, OH 45439

*COMPUTE!*  
P.O. Box 5406  
Greensboro, NC 26403

*COMPUTER BASICS (Newsletter)*  
*For Non-Computer Managers and Professionals*  
Information Plus  
1165 Long Hill Road  
Sterling, NJ 07980

*COMPUTERWORLD*  
CW Communications, Inc.  
Box 880  
375 Cochituate Road  
Framingham, MA 01701

*THE COMPUTING TEACHER*

International Council for Computers in  
Education  
1787 Agate Street  
University of Oregon  
Eugene, OR 97403

*CONSUMER REPORTS*

Consumers Union of United States, Inc.  
256 Washington Street  
Mount Vernon, NY 10553

*CREATIVE COMPUTING: The Magazine of  
Recreational and Educational Computing*

P.O. Box 789-M  
Morristown, NJ 07968

*DATA BASED ADVISOR*

Data Base Solutions  
1975 Fifth Avenue  
Suite 105  
San Diego, CA 92101

*DATA COMMUNICATIONS*

McGraw-Hill, Inc. 1221 Avenue of the  
Americas  
New York, NY 10020

*DATAMATION*

Cahners Publishing Co.  
249 West 17th St.  
New York, NY 10011

*DESKTOP COMPUTING*

Wayne Green, Inc.  
80 Pine Street  
Peterborough, NH 03458

*EDUCATIONAL TECHNOLOGY*

Educational Technology Publications, Inc.  
720 Palisade Avenue  
Englewood Cliffs, NJ 07632

*ELECTRONIC LEARNING*

Scholastic Learning, Inc.  
730 Broadway  
New York, NY 10003

*ELECTRONIC MAIL & MESSAGE  
SYSTEMS*

International Resource Development, Inc.  
30 High Street  
Norwalk, CT 16851

*ELECTRONIC OFFICE MANAGEMENT  
& TECHNOLOGY: MICRO WORLD  
TELEPHONE EQUIPMENT SELECTION  
GUIDE*

Auerbach Publishers  
6560 North Park Drive  
Pennsauken, NJ 08109

*THE ERGONOMIC NEWSLETTER*

3029 Wilshire Boulevard, Suite 200  
Santa Monica, CA 90403

*FAMILY COMPUTING*

Scholastic, Inc.  
351 Garver Rd.  
P.O. Box 2700  
Monroe, OH 45050-2700

*FORTUNE*

Time, Inc.  
Time-Life Building  
1271 Avenue of the Americas  
New York, NY 10020

*THE FUTURIST*

Publication of The World Future Society  
4916 St. Elmo Avenue  
Washington, DC 20814

*HIGH TECHNOLOGY*

Technology Publishing Company  
38 Commercial Wharf  
Boston, MA 02110

*ICP SOFTWARE BUSINESS REVIEW*

International Computer Programs, Inc.  
9000 Keystone Crossing  
P.O. Box 40946  
Indianapolis, IN 46240

*IMC JOURNAL*

International Information Management  
Congress  
P.O. Box 34404  
Bethesda, MD 20817

*IMPACT: INFORMATION TECHNOLOGY*

Administrative Management Society  
2360 Maryland Road  
Willow Grove, PA 19090

Business Computer Programming I  
APPENDIX A: Reference Addresses

*INFORMATION MANAGEMENT*

Systems Technology Software Media  
PTN Publishing Company  
101 Crossways Park West  
Woodbury, NY 11797

*INFORMATION SYSTEMS NEWS*

CMP Publications  
111 East Shore Road  
Manhasset, NY 11030

*INFOSYSTEMS*

Hitchcock Publishing Company  
Hitchcock Building  
25 West 550 Geneva Road  
Wheaton, IL 60188

*INFOWORLD*

Popular Computing, Inc.  
1060 Marsh Road,  
Suite C-200  
Menlo Park, CA 94025

*INTERFACE AGE: MICROCOMPUTING  
FOR HOME AND SMALL BUSINESS*

13913 Artesia Blvd.  
Cerritos, CA 90701

*KILOBAUD: THE SMALL COMPUTER  
MAGAZINE*

Peterborough, NH 03458

*MAC WORLD*

PC World Communications, Inc.  
555 DeHaro Street  
San Francisco, CA 94107

*MANAGEMENT TECHNOLOGY*

International Thomson Technology Informa-  
tion, Inc.  
135 West 50 Street  
New York, NY 10020

*MODERN OFFICE TECHNOLOGY*

Penton/IPC, Inc.  
Penton Plaza  
1111 Chester Avenue  
Cleveland, OH 44114

*NATION'S BUSINESS*

Chamber of Commerce of the United States  
1615 H Street, N.W.  
Washington, DC 20062

*NEWSWEEK*

Newsweek, Inc.  
444 Madison Avenue  
New York, NY 10022

*THE OFFICE*

Office Publications, Inc.  
1200 Summer Street  
Stamford, CT 06904

*OFFICE ADMINISTRATION AND  
AUTOMATION*

Dalton Communications, Inc.  
1123 Broadway  
New York, NY 10010

*THE OFFICE PROFESSIONAL (Newsletter)*

1316 Sam Bass Circle  
Round Rock, TX 78664

*OFFICE SYSTEMS ERGONOMICS REPORT*

The Koffler Group  
3029 Wilshire Boulevard,  
Suite 200  
Santa Monica, CA 90403

*OMNI*

Omni Publications International, Ltd.  
1965 Broadway  
New York, NY 10023-5965

*PC: AN INDEPENDENT GUIDE TO IBM  
PERSONAL COMPUTERS*

Ziff-Davis Publishing Company  
One Park Avenue  
New York, NY 10016

*PC MAGAZINE*

39 East Hanover Avenue  
Morris Plains, NJ 07950

*PC WEEK (Weekly Magazine)*

The National Newspaper of  
IBM Standard Microcomputing  
Ziff-Davis Publishing Company  
381 Elliot Street  
Newton, MA 02164

*PC WORLD*

555 DeHaro Street  
San Francisco, CA 94107

*PERSONAL COMPUTING*

50 Essex Street  
Rochelle Park, NJ 07662

**Business Computer Programming I**  
**APPENDIX A: Reference Addresses**

**PERSONAL COMPUTING PLUS**

Hayden Publishing Company, Inc.  
10 Mulholland Drive  
Hasbrouck Heights, NJ 07604

**PERSONAL PUBLISHING**

Box 390  
Itasca, IL 60143

**POPULAR COMPUTING**

70 Main Street  
Peterborough, NH 03458

**POPULAR SCIENCE**

Times-Mirror Magazines, Inc.  
Communications Department  
380 Madison Avenue  
New York, NY 10017

**RUN: THE COMMODORE 64/128 USERS GUIDE**

C. W. Communications  
P.O. Box 954  
Farmingdale, NY 11737

**SAVVY**

Savvy Company  
111 Eighth Avenue  
New York, NY 10011

**SCIENCE DIGEST**

Hearst Publications, Science Digest  
888 Seventh Avenue  
New York, NY 10619

**THE SECRETARY**

Professional Secretaries International  
301 East Armour Boulevard  
Kansas City, MO 64111

**SOFTWARE DIGEST**

EDP Services, Inc.  
7620 Turtle Turnpike  
Annadale, VA

**SOFTWARE REVIEW**

Microform Review, Inc.  
520 Riverside Drive  
Westport, CT 06880

**THE TECHNOLOGICAL HORIZONS IN EDUCATION JOURNAL**

Information Synergy, Inc.  
2626 South Pullman  
Santa Ana, CA 92705

**TECHNOLOGY ILLUSTRATED**

Technology Publishing Company  
38 Commercial Wharf  
Boston, MA 02110

**TEXAS COMPUTING**

17806 Davenport,  
Suite 105  
Dallas, TX 75252  
(214)931-0157

**TIME**

Time, Inc.  
Time-Life Building  
1271 Avenue of the Americas  
New York, NY 10020

**TODAY'S OFFICE**

United Technical Publications, Inc.  
645 Stewart Avenue  
Garden City, NY 11530

**TRAINING & DEVELOPMENTAL JOURNAL**

American Society for Training and Development (available by membership only)  
600 Maryland Avenue, S.W., Suite 305  
Washington, DC 20024

**WORDS MAGAZINE**

Association of Information Systems Professionals  
1015 North York Road  
Willow Grove, PA 19090

**WORKING WOMAN**

Hal Publications, Inc.  
342 Madison Avenue  
New York, NY 10173

**SERVICES AND REPORTS**

**DATAPRO REPORTS ON MICROCOMPUTERS**

Datapro Research Corporation  
1805 Underwood Boulevard  
Delran, NJ 08075

**DATAPRO REPORTS ON MICROCOMPUTER MANAGEMENT**

Datapro Research Corporation  
1805 Underwood Boulevard  
Delran, NJ 08075

**Business Computer Programming I**  
**APPENDIX A: Reference Addresses**

***DATAPRO REPORTS ON  
MICROCOMPUTER SOFTWARE***

Datapro Research Corporation  
1805 Underwood Boulevard  
Delran, NJ 08075

***DATAPRO REPORTS ON WORD  
PROCESSING***

Datapro Research Corporation  
1805 Underwood Boulevard  
Delran, NJ 08075

***THE SEYBOLD REPORT ON OFFICE  
SYSTEMS***

Seybold Publications, Inc.  
Box 644  
Media, PA 19063

**PROFESSIONAL ORGANIZATIONS**

**ADMINISTRATIVE MANAGEMENT  
SOCIETY**

Maryland Road  
Willow Grove, PA 19090  
(215)659-4300

**AMERICAN FEDERATION OF INFORM-  
ATION PROCESSING SOCIETIES, INC.**

1815 North Lynn Street  
Arlington, VA 22209

**AMERICAN MANAGEMENT  
ASSOCIATIONS**

135 West 50 Street  
New York, NY 10020  
(212)586-8100

**AMERICAN SOCIETY FOR  
INFORMATION SCIENCE**

1155 16th Street, N.W.  
Washington, D.C. 20036  
(202)659-3644

**AMERICAN SOCIETY FOR TRAINING  
AND DEVELOPMENT**

P.O. Box 5307  
Madison, WI 53705

**AMERICAN VOCATIONAL  
ASSOCIATION**

2020 North 14 Street  
Arlington, VA 22201  
(703)522-6121

**ASSOCIATION OF COMPUTING  
MACHINERY**

11 West 42 Street  
New York, NY 10036  
(212)869-7440

**ASSOCIATION FOR COMPUTER  
USERS (ACU)**

1133 Avenue of the Americas  
New York, NY 10036  
(212)265-6309

**ASSOCIATION FOR EDUCATIONAL  
COMMUNICATIONS AND  
TECHNOLOGY (AECT)**

1126 16th Street, N.W.  
Washington, D.C. 20036  
(202) 833-4180

**ASSOCIATION FOR EDUCATIONAL  
DATA SYSTEMS (AEDS)**

1201 16th Street, N.W.  
Washington, D.C. 20036  
(202)833-4100

**ASSOCIATION FOR THE  
DEVELOPMENT OF COMPUTER-BASED  
INSTRUCTIONAL SYSTEMS (ADCIS)**

Bond Hall  
Western Washington University  
CF Computer Center  
Bellingham, WA 98225  
(206)676-2860

**ASSOCIATION OF INFORMATION  
SYSTEMS PROFESSIONALS**

1015 North York Road  
Willow Grove, PA 19090  
(215)657-6300

**ASSOCIATION OF RECORDS  
MANAGERS AND ADMINISTRATORS**

4200 Somerset, Suite 215  
Prairie Village, KS 66208  
(913)341-3808

**COMPUTER-BASED EDUCATION  
RESEARCH LABORATORY**

University of Illinois  
Urbana, IL 61801

**COMPUTER-USING EDUCATORS**

P.O. Box 18547  
San Jose, CA 95158



**Business Computer Programming I**  
**APPENDIX A: Reference Addresses**

**COUNCIL FOR EDUCATIONAL  
DEVELOPMENT AND RESEARCH**  
1518 K Street, N.W.  
Suite 206  
Washington, D.C.

**DATAPRO RESEARCH COMPANY**  
1805 Underwood Blvd.  
Delran, NJ 08075  
(800)257-9406

**EDUCATIONAL TECHNOLOGY  
CENTER**  
University of California  
Irvine, CA 92717  
(714)833-6911

**EDUCOM**  
P.O. Box 364  
Princeton, NJ 08540  
(609)734-1915

**INTERNATIONAL COUNCIL FOR  
COMPUTERS IN EDUCATION (ICCE)**  
Department of Computer and Information  
Science  
University of Oregon  
1787 Agate Street  
Eugene, OR 97403  
(503)686-4408

**LABORATORY FOR PERSONAL  
COMPUTERS IN EDUCATION**  
State University of New York  
Stony Brook, NY 11094  
(516)246-8418

**MICROSOFT**  
Northwest Regional Educational Laboratory  
500 Lindsay Building  
710 2nd Avenue, S.W.  
Portland, OR 97204  
(503)248-6974

**MICROCOMPUTER CENTER**  
San Mateo Educational Resources Center  
Library  
333 Main Street  
Redwood City, CA 94063  
(415)363-5469

**MICROCOMPUTER EDUCATION  
APPLICATIONS NETWORK**  
256 North Washington Street  
Falls Church, VA 22046

**MICROCOMPUTER RESOURCE  
CENTER**  
Teachers College  
Columbia University  
New York, NY 10027  
(212)678-3740

**MINNESOTA EDUCATIONAL  
COMPUTING CONSORTIUM (MECC)**  
2520 Broadway Drive  
St. Paul, MN 55113  
(612)376-1101

**NATIONAL ASSOCIATION OF  
COMPUTER STORES**  
3255 South U.S.1  
Fort Pierce, FL 33450  
(305)465-9450

**NATIONAL BUSINESS EDUCATION  
ASSOCIATION**  
1914 Association Drive  
Reston, VA 22091  
(703)860-8300

**NATIONAL EDUCATION ASSOCIATION**  
1201 16th Street, N.W.  
Washington, D.C. 20036

**NINE TO FIVE NATIONAL  
ASSOCIATION OF WORKING WOMEN**  
1224 Huron Road  
Cleveland, OH 44115  
(216)566-9308

**OFFICE TECHNOLOGY MANAGEMENT  
ASSOCIATION**  
9401 West Beloit Road, Suite 101  
Milwaukee, WI 53227  
(414)321-0880

**PROFESSIONAL SECRETARIES  
INTERNATIONAL**  
301 East Armour Boulevard  
Kansas City, MO 64111  
(816)531-7010

**TEXAS BUSINESS EDUCATION ASSOCIATION**  
Bette Carpenter, Executive Secretary  
Amarillo, TX 79109

**TEXAS COMPUTER EDUCATION ASSOCIATION**  
7131 Midbury  
Dallas, TX 75230  
(214)361-9472

### **BUSINESS JOURNALS**

**BUSINESS EDUCATION FORUM**  
National Business Education Assn.  
1914 Association Drive  
Reston, VA 22091  
The journal features articles for teachers of secretarial programs, typewriting, accounting, clerical programs, and basic business, as well as economics, marketing and distribution, communication, and data processing.

**BUSINESS EDUCATION JOURNAL**  
Emporia State University  
Division of Business and Business Education  
1200 Commercial St.  
Emporia, KS 66801  
The journal contains general business topics with emphasis on business production and teacher education.

**BUSINESS EDUCATION WORLD**  
Gregg Division  
McGraw-Hill Book Company  
1221 Avenue of the Americas  
New York, NY 10020  
Innovative application of teaching methodology, educational concepts, descriptions of new curricula and trends in business and office education subjects. Free to business teachers.

**BUSINESS HORIZONS**  
Graduate School of Business  
Indiana University  
Bloomington, IN 47405  
Articles by academics and business people on significant business issues, trends, and research projects.

**DELTA PI EPSILON JOURNAL**  
Delta Pi Epsilon National Office  
Gustovus Adolphus College  
St. Peter, MN 56082  
Devoted to research articles dealing with theory, practice, and research in business education.

**JOURNAL OF BUSINESS EDUCATION**  
Heldref Publications  
4000 Albermarle St. NW  
Washington, DC 20016  
A journal dealing with business education from high school to community college level. Contributors are principally teachers and academics.

**VOCED**  
formerly, AMERICAN VOCATIONAL JOURNAL  
2020 N. 14th Street  
Arlington, VA 22201  
The journal of the American Vocational Association published for teachers, administrators, counselors, and laypersons involved in the task of preparing people for work. The journal gives a balanced view of both the theoretical and practical aspects of vocational education.

### **OTHER BUSINESS PUBLICATIONS**

**THE BALANCE SHEET**  
South-Western Publishing Co.  
5101 Madison Rd.  
Cincinnati, OH 45227  
This publication is free to teachers, and features articles of current interest relating to business and economic course offerings at the high school and postsecondary levels.

**MODERN OFFICE PROCEDURES MAGAZINES**  
Pent/IPC  
1111 Chester Avenue  
Cleveland, OH 44114  
Business/office market magazine featuring office procedures and utilization of new technological developments in computers, business machines, furnishings, and supplies.

*THE OFFICE*

Office Publications, Inc.  
1200 Summer Street  
Stamford, CT 06905

The magazine discusses many aspects of office practice and management for both executives and office managers. Articles include topics such as systems analysis, programming, employee relations, applications of computers, and product/equipment announcements.

*OFFICE GUIDE*

Bureau of Business Practices  
24 Rope Ferry Road  
Waterford, CT 06385-9985

Issued twice each month, the six-page publication offers office hints and shortcuts that can be used as a training tool to upgrade office personnel.

*OFFICE GUIDE FOR WORKING WOMEN*

Bureau of Business Practices  
24 Rope Ferry Road  
Waterford, CT 06385-9985

The magazine offers general articles and regular features on typing, vocabulary, and grammar applicable to office workers and/or secretarial science students.

*THE OFFICE PROFESSIONAL*

1316 Sam Bass Circle  
Round Rock, TX 78664

Provides reliable information about office systems and procedures, secretarial methods, management thinking, and office technology. Also features articles on communication techniques, professional human relations skills, and the latest approaches to career development.

*OFFICE TALKS*

Bureau of Business Practices  
24 Rope Ferry Road  
Waterford, CT 06385-9985

Featuring office techniques, personal appearance, office etiquette, cosmetics, and hygiene for secretaries.

*OPEN ENTRIES*

The Center for Studies in  
Vocational Education  
Florida State University  
Stone Building  
Tallahassee, FL 32306

A nationwide communication link for educators involved in competency-based vocational education that encourages exchange of information on competency-based learning materials, development of new guides, packets, audiovisual materials, and computer-based resources.

*P.S. FOR PROFESSIONAL SECRETARIES*

Bureau of Business Practices  
24 Rope Ferry Road  
Waterford, CT 06385-9985

Helps office personnel to perform their tasks more quickly, easily, and efficiently.

*THE SECRETARY*

Professional Secretaries International  
301 East Armour Blvd.  
Kansas City, MO 64111-1299

A publication for professional secretaries featuring articles by management, business educators, and secretaries that stresses improving knowledge and ability in secretarial expertise.

*TEXAS BUSINESS EDUCATOR*

Texas Business Educational Assn.  
3500 Kileen  
Amarillo, TX 79109

A semiannual bulletin of the activities of the Texas Business Education Association, plus articles of current interest to business educators.

*TODAY'S OFFICE*

Hearst Business Communications  
UTP Division  
645 Stewart Avenue  
Garden City, NY 11530

A professional business magazine covering the office.

*VIEWPOINTS IN BUSINESS EDUCATION*

Milady Publishing Corp.  
3839 White Plains Road  
Bronx, NY 10467

A teaching methods and materials newsletter for business education teachers.



**Business Computer Programming I**  
**APPENDIX A: Reference Addresses**

**MID-AMERICA VOCATIONAL  
CURRICULUM CONSORTIUM  
PUBLICATIONS**

**BASIC PROGRAMMING**

**INFORMATION PROCESSING**

**INTRODUCTION TO MICROCOM-  
PUTER APPLICATIONS**

**INTRODUCTION TO TELECOM-  
MUNICATIONS**

**MICROCOMPUTER BUSINESS  
APPLICATION**

To order MAVCC publications, contact:

Educational Development & Training Center  
East Texas State University  
Commerce, TX 75429  
(214)886-5623

689

## COURSE DESCRIPTIONS AND CODES

**Microcomputer Applications\*** 11673100  
Grades 10-12, 1/2-1 unit

A cluster course which introduces basic concepts and provides hands-on skills related to microcomputer systems and applications. Special emphasis is placed on basic skills for the following: word processing, database, spreadsheet, telecommunications, desktop publishing, electronic image transfer, networking, integrated software, and social implications of using computers. This course is designed as a foundation course and provides preparation for advancement into the occupationally specific courses.

**Office Support Systems** 11673000  
Grades 10-12, 1/2-1 unit

A cluster course which introduces the basic concepts and skills needed for general office positions. Instruction and skill development are provided in keyboarding, filing, financial activities, office machines, and securing and maintaining employment. This course provides entry-level employment skills for the office or for advancement into an occupationally specific course.

**Business Financial Applications** 11673200  
Grades 10-12, 1/2-1 unit

A cluster course which introduces the basic concepts and skills necessary for employment in financial institutions. Special emphasis will be placed on banking operations, credit administration, types of loans, legal aspects of banking, and other financial institutions. This course provides entry-level information for working in a financial institution or for advancement into an occupationally specific course.

**Business Information Processing\*** 11672100 (2+2)  
Grades 11-12, 1-3 units (each year the course is taught)  
Preemployment Laboratory or Cooperative Education

An occupationally specific course which introduces the concepts and skills for producing business documents and correspondence through the use of the document cycle and word processing equipment. Special emphasis is placed on automated management of business records, database management, and electronic spreadsheets. Management of information flow, electronic mail, desktop publishing, electronic bulletin boards, graphics, and on-line information services are also included. The course provides preparation for employment or advanced standing at the postsecondary level.

**Office Administrative Systems** 11672000 (2+2)  
Grades 11-12, 1-3 units  
Preemployment Laboratory or Cooperative Education

An occupationally specific course which includes advanced concepts and skills necessary for a general office position through a simulated office environment. Special emphasis is placed on keyboarding, records management, communications, operation of office machines, and the electronic production of business documents. The course prepares the student for office employment or advanced standing at the postsecondary level.

**Business Computer Programming I**  
**APPENDIX B: Course Descriptions**

**Business Computer Applications I\* 1167220 (2+2)**

**Grades 11-12, 1-3 units ((each year the course is taught)**

**Preemployment Laboratory or Cooperative Education**

An occupationally specific course which introduces the advanced concepts and skills related to computer applications. Special emphasis is placed on basic computer operations, word processing, database management, spreadsheet manipulation, telecommunications, desktop publishing, and other high-level business application software. This course is designed to provide the foundation for more advanced computer applications designated in Business Computer Applications II or advanced standing at the postsecondary level.

**Business Computer Applications II\* 11672300**

**Grades 11-12, 1-3 units**

**Preemployment Laboratory or Cooperative Education**

An occupationally specific course which includes advanced concepts and skills related to advanced computer applications. Special emphasis is placed on teleprocessing, integrated software systems, computer-generated graphics, algorithmic approaches to problem solving, and introductory programming skills. This course provides preparation for employment in a business computer occupation or advanced standing at the postsecondary level.

**Business Computer Programming I\* 11672400 (2+2)**

**Grades 11-12, 1-3 units**

**Preemployment Laboratory or Cooperative Education**

An occupationally specific course which emphasizes concepts and skills related to business computer programming. Special emphasis is placed on programming concepts and languages, programming concepts and languages, programming methodology, skills related to data storage, and an awareness of programming applications for business. This course is designed to lay a foundation for further occupationally specific business programming instruction or advanced standing at the postsecondary level.

**Business Computer Programming II\* 11672500**

**Grades 11-12, 1-3 units**

**Preemployment Laboratory or Cooperative Education**

An occupationally specific course which includes advanced concepts and skills related to business computer programming. Special emphasis is placed on bringing development efforts to full production status in environments, and on languages commonly used in development of commercial software for business applications. Concepts such as design, maintenance, testing, team development, and user documentation are included. This course prepares students for employment in a computer-related occupation or advanced placement at the postsecondary level.

**Business Office Services\*\*, CVAE--12670500, VEH--13670500**

**Grades 9-12, 1-3 units, CVAE**

**Grades 7-12, 1-3 units, VEH**

A course which introduces basic concepts and skills needed for a general office position with concentration on the use of reprographic equipment, keyboarding, filing, and office machines. Special emphasis is placed on remediation in the areas of basic math and English usage, as well as securing and maintaining employment.

Business Computer Programming I  
APPENDIX B: Course Descriptions

**Business Office Clerical\*\*, CVAE--12670600, VEH--13670400**  
**Grades 9-12, 1-3 units, CVAE**  
**Grades 7-12, 1-3 units, VEH •**

A course which introduces basic concepts and skills for a general office position with concentration on the use of keyboarding, filing, office machines, communications, and reprographics. Special emphasis is placed on remediation in the areas of basic math and English usage, as well as securing and maintaining employment.

- \* Recommended for approval to satisfy the computer science requirement for the advanced high school program.
- \*\* Students may enroll each year the course is offered

**COMMON ESSENTIAL ELEMENTS**  
**for all**  
**OFFICE EDUCATION PROGRAMS**

**75.87 Office Education**

- (a) The elements in this subsection are common to office education programs and shall be included in each course at the appropriate level. They are described here to preclude repetition in each course. Every school offering office education shall provide courses which include the following essential elements:

- (1) Leadership concepts and skills. The student shall be provided opportunities to:
  - (A) demonstrate skills, characteristics, and responsibilities of leaders and effective group members;
  - (B) demonstrate a knowledge of parliamentary procedure principles;
  - (C) plan and conduct leadership activities; and
  - (D) prepare for effective citizenship and for participation in our democratic society.
- (2) Concepts and skills related to successful employment and/or postsecondary training. The student shall be provided opportunities to:
  - (A) identify employment opportunities and preparation requirements in the chosen field;
  - (B) identify effective methods to secure and terminate employment;
  - (C) demonstrate effective communication skills both oral and written, and follow through on assigned tasks;
  - (D) demonstrate dependability and punctuality;
  - (E) demonstrate productive work habits and attitudes;
  - (F) understand the importance of taking pride in the quality of work performed;
  - (G) recognize the dignity in work;
  - (H) develop skills in planning and organizing work;
  - (I) apply required methods and sequences when performing tasks;
  - (J) apply principles of time management and work simplification when performing assigned tasks;

- (K) identify ethical practices and responsibilities; and
  - (L) understand the importance of the application of organizational policies and procedures.
- (3) Concepts and skills associated with entrepreneurship. The student shall be provided opportunities to:
- (A) identify opportunities for business ownership;
  - (B) understand the risk and profit motive factor;
  - (C) understand the elements and advantages of the free enterprise system; and
  - (D) explain the role of small business in the free enterprise system.
- (4) Concepts and skills related to safety and safe working conditions. The student shall be provided opportunities to identify and apply safe working practices to all training situations.
- (5) Concepts and skills associated with human relations and personality development. The student shall be provided opportunities to:
- (A) understand the importance of maintaining good health and proper appearance for effective job performance;
  - (B) understand oneself and others;
  - (C) exercise self-control;
  - (D) accept and use criticism;
  - (E) recognize basic human relationships as they relate to business success; and
  - (F) demonstrate characteristics for successful working relationships.
- (6) Concepts and skills related to personal and business management. The student shall be provided opportunities to:
- (A) explain how management assists in reaching personal and family goals;
  - (B) explain the management process;
  - (C) describe the role of management in controlling stress;
  - (D) identify and understand personal checking accounts;
  - (E) identify and understand personal loan application processes;
  - (F) identify and understand different financial institutions;

**Business Computer Programming I**  
**APPENDIX C: Office Education Essential Elements**

- (G) identify the role and functions of business management;
- (H) understand the lines of authority; and
- (I) identify effective supervisory techniques.



OFFICE EDUCATION ESSENTIAL ELEMENTS  
**OFFICE ADMINISTRATIVE SYSTEMS**  
(Blended Course)

75.87

- (b) **Office Administrative Systems (1, 2, or 3 units) shall include the following essential elements:**
- (1) Concepts and skills related to office administration. The student shall be provided opportunities to:
    - (A) identify duties and responsibilities common to most offices;
    - (B) delineate the workflow of an office or system;
    - (C) relate office operations to the entire business organization;
    - (D) establish priorities and develop organizational abilities;
    - (E) interpret and apply information from resource materials; and
    - (F) identify and apply concepts and skills related to safety and safe working conditions in the office environment.
  - (2) Concepts and skills related to successful employment and/or postsecondary training. The student shall be provided opportunities to:
    - (A) identify employment opportunities and preparation requirements in the chosen field; and
    - (B) identify effective methods to secure, maintain, and terminate employment.
  - (3) Concepts and skills necessary for effective communications in the office environment. The student shall be provided opportunities to:
    - (A) identify proper techniques in business telephone usage;
    - (B) prepare effective communication using the document production cycle;
    - (C) identify and use proper oral and written language skills relating to business;
    - (D) identify current terminology relating to business and technology; and
    - (E) identify classes of mail, special mailing services, and procedures of handling incoming and outgoing mail.



- (4) Concepts and skills related to information management and control. The student shall be provided opportunities to:
  - (A) identify equipment and supplies necessary for a filing system;
  - (B) identify correct usage in applying terminology and procedures for indexing, coding, and filing using the alphabetic, numeric, subject, and geographic system;
  - (C) identify, describe, and/or demonstrate procedures for controlling manual records and computerized information; and
  - (D) identify a transfer and retention schedule.
- (5) Concepts and skills necessary for the operation of office equipment. The student shall be provided opportunities to:
  - (A) use the touch system to produce accurate mailable business documents using electronic equipment;
  - (B) demonstrate the ability to enter and retrieve data using a computer;
  - (C) use the touch system on the electronic calculator/computer to compute business problems;
  - (D) identify appropriate reprographic processes for specific tasks; and
  - (E) demonstrate ability to produce legible copies.
- (6) Concepts and skills associated with the social implications of computers. The student shall be provided opportunities to:
  - (A) examine issues concerning responsible use of computer systems; and
  - (B) understand the social ramifications of computer applications related to privacy, values, and reliability of systems.

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**OFFICE ADMINISTRATIVE SYSTEMS II**

*(State Board Approval Pending Fall 1990)*

75.87

*Office Administrative Systems II (1, 2, or 3 units) shall include the following essential elements:*

- (1) Advanced concepts and skill simulation related to office administration. The student shall be provided opportunities to:*
  - (A) perform duties and responsibilities common to most offices;*
  - (B) establish the workflow of an office or system;*
  - (C) relate office operations to the entire business organization;*
  - (D) establish priorities and develop organizational abilities; and*
  - (E) interpret data, make decisions, and apply information from resource materials.*
- (2) Advanced concepts and skills necessary for effective communications in the office environment. The student shall be provided opportunities to:*
  - (A) apply proper techniques in business telephone usage;*
  - (B) identify the use of teleprocessing systems for data exchange in an actual business setting;*
  - (C) compose and produce effective communication using the document production cycle;*
  - (D) use proper oral and written language skills relating to business; and*
  - (E) identify current terminology relating to business and technology.*
- (3) Advanced concepts and skills related to information management and control. The student shall be provided opportunities to:*
  - (A) demonstrate procedures for organizing and filing materials;*
  - (B) identify correct usage in applying terminology and procedures for indexing, coding, and filing using the alphabetic, numeric, subject, and geographic system; and*

- (C) *demonstrate procedures for accessing and maintaining computerized information.*
- (4) *Advanced concepts and skills simulation for the operation of office equipment. The student shall be provided opportunities to:*
  - (A) *demonstrate keyboarding speed and accuracy necessary for entry-level employment;*
  - (B) *produce mailable business documents using microcomputers;*
  - (C) *demonstrate the ability to enter and retrieve data using a computer;*
  - (D) *use the touch system on the electronic calculator/computer to compute business problems; and*
  - (E) *utilize appropriate reprographic processes for specific tasks, including desktop publishing.*

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**BUSINESS INFORMATION PROCESSING**

75.87

- (c) **Business Information Processing (1, 2, or 3 units) shall include the following essential elements:**
- (1) Concepts and skills related to effective communication and production of business documents. The student shall be provided opportunities to:
    - (A) demonstrate proficiency in business English, spelling, and proofreading;
    - (B) identify and apply correct format for business correspondence and documents;
    - (C) demonstrate basic information processing concepts and processes to employ the appropriate steps in document production;
    - (D) demonstrate proficiency in transcription;
    - (E) identify and use computer-based tools for document production;
    - (F) demonstrate basic knowledge and usage of business machines; and
    - (G) identify proper techniques in business telephone usage.
  - (2) Concepts and skills related to business records management. The student shall be provided opportunities to:
    - (A) identify procedures for controlling manual records by applying terminology and procedures for indexing, coding, and filing alphabetically;
    - (B) identify and apply procedures for automated management of business records, at least including use of database management software for storage, retrieval, manipulation, selection, and reporting of records;
    - (C) identify business applications of electronic spreadsheets and understand and/or demonstrate fundamental concepts and operational procedures related to their use.
  - (3) Concepts and skills related to management of information flow out of, into, and within an office. The student shall be provided opportunities to:
    - (A) understand concepts related to computer-aided preparation and distribution of personalized correspondence, at least including use of mail merge and electronic mail.

- (B) identify tools for preparing information for mass distribution, at least including desktop publishing, production of presentation graphics, and maintenance of electronic bulletin boards;
  - (C) understand the concepts of using communication systems both for accessing remotely stored information (at least including use of on-line information services) and for transmission of outgoing information; and
  - (D) understand concepts related to transporting of images electronically.
- (4) Identify characteristics of contemporary office systems related to information processing and management. The student shall be given opportunities to:
- (A) observe office activity related to information processing and information management;
  - (B) discuss responsibilities and procedures involved in business information processing with individuals working in the field;
  - (C) read and discuss current literature related to business information processing; and
  - (D) identify emerging information processing tools and understand their impact on business information processing.

OFFICE EDUCATION ESSENTIAL ELEMENTS

**BUSINESS INFORMATION PROCESSING II**

(State Board Approval Pending Fall 1990)

75.87

*Business Information Processing II (1, 2, or 3 units) shall include the following elements:*

- (1) *Concepts and skills related to effective communication and production of business documents. The students shall be provided opportunities to:*
  - (A) *demonstrate proficiency in business English, spelling, proofreading in production of original business documents, and/or editing work previously created;*
  - (B) *produce documents through the use of text manipulation and/or high-level integrated software; and*
  - (C) *compose, dictate, and transcribe business documents.*
- (2) *Concepts and skills related to business information management. The students shall be provided opportunities to apply decision making skills:*
  - (A) *using advanced procedures and applications for automated management of business information using a database management system.*
  - (B) *using advanced applications of electronic spreadsheets.*
- (3) *Concepts and skills related to management of information flow out of, into, and within an office. The students shall be provided opportunities to:*
  - (A) *integrate word processing, database, and spreadsheet software using import/export commands and/or ASCII files;*
  - (B) *understand the process for setup installation and use of software by reading documentation prepared by the software manufacturer;*
  - (C) *identify and apply concepts of layout and design in desktop publishing, including graphics, fonts, and paper for final copy;*
  - (D) *demonstrate proficiency in the use of presentation graphics to aid in presentation of an oral report;*
  - (E) *identify criteria for selection and evaluation of software; and*
  - (F) *apply concepts related to transmitting information electronically.*

OFFICE EDUCATION ESSENTIAL ELEMENTS

**BUSINESS OFFICE SERVICES**

75.87

- (d) **Coordinated Vocational Academic Education (CVAE) Business Office Services (1-2 units) shall include the following essential elements:**
- (1) Concepts and skills involved in keyboarding. The student shall be provided opportunities to:
    - (A) demonstrate the skills involved in keyboarding;
    - (B) apply appropriate procedures for proofreading and correcting errors; and
    - (C) demonstrate and identify information processing concepts in keyboarding.
  - (2) Concepts and skills necessary for filing. The student shall be provided opportunities to:
    - (A) demonstrate procedures used in alphabetic and numeric filing; and
    - (B) identify procedures used in electronic filing.
  - (3) Concepts and skills related to the operation of office equipment. The student shall be provided opportunities to:
    - (A) identify the procedures and equipment used in completing layouts for reprographics;
    - (B) identify the procedures used in operating reprographic equipment;
    - (C) demonstrate the ability to prepare masters and to operate the offset press;
    - (D) demonstrate the ability to use desktop publishing features;
    - (E) demonstrate the ability to solve mathematical problems using the touch method on the electronic calculator/computer keypad; and
    - (F) demonstrate the ability to enter and retrieve data on a microcomputer.
  - (4) Concepts and skills necessary for processing mail. The student shall be provided opportunities to:
    - (A) identify the procedures involved in processing incoming and outgoing mail; and
    - (B) identify the procedures involved in transmitting mail electronically.



- (5) Concepts and skills necessary for effective communications. The student shall be provided opportunities to:
  - (A) develop effective listening skills;
  - (B) identify and use proper business techniques when using the telephone;
  - (C) prepare effective written communications; and
  - (D) identify and use proper language usage skills in business.
- (6) Concepts and skills necessary for securing and maintaining employment. The student shall be provided opportunities to:
  - (A) demonstrate the skills necessary for job application and interview; and
  - (B) apply human relations skills and the ability to work with others.
- (7) Concepts and skills associated with the social implications of computers. The student shall be provided opportunities to:
  - (A) examine issues concerning responsible use of computer systems;
  - (B) understand the social ramifications of computer applications related to privacy, values, and reliability of systems.



OFFICE EDUCATION ESSENTIAL ELEMENTS  
**BUSINESS COMPUTER APPLICATIONS I**  
(Blended Course)

75.87

- (e) **Business Computer Applications I (1, 2, or 3 units) shall include the following essential elements:**
- (1) Concepts and skills related to computer operations. The student shall be provided opportunities to:
    - (A) demonstrate comprehension of computer-related vocabulary;
    - (B) describe the functions of the components of a computer system;
    - (C) demonstrate and apply the capabilities of an operating system; and
    - (D) explore the concepts of telecommunication techniques and networking.
  - (2) Concepts and skills related to word processing. The student shall be provided opportunities to:
    - (A) demonstrate word processing concepts;
    - (B) apply higher-level capabilities of word processing tools and software; and
    - (C) understand the procedures and recognize the importance of merging data processing and word processing.
  - (3) Concepts and skills related to spreadsheets. The student shall be provided opportunities to:
    - (A) demonstrate higher-level spreadsheet concepts; and
    - (B) apply higher-level capabilities of spreadsheets.
  - (4) concepts and skills related to database management. The student shall be provided opportunities to:
    - (A) demonstrate database concepts, procedures, and capabilities; and
    - (B) apply higher-level capabilities of database management.

- (5) Concepts and skills related to advanced application software. The student shall be provided opportunities to:
  - (A) discuss integration of people and computers for productivity;
  - (B) demonstrate the ability to apply information processing concepts and skills to realistic business situations;
  - (C) demonstrate the ability to apply information processing concepts and skills to realistic personal situations; and
  - (D) identify tools for preparing information for mass distribution including desktop publishing, graphics, and maintenance of electronic bulletin boards.
- (6) Concepts and skills associated with the social implications of computers. The student shall be provided opportunities to
  - (A) examine issues concerning responsible use of computer systems; and
  - (B) understand the social ramifications of computer applications related to privacy, values, and reliability of systems.

766

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**BUSINESS COMPUTER APPLICATIONS II**

75.87

- (f) **Business Computer Applications II (1, 2, or 3 units) shall include the following essential elements:**
- (1) Concepts and skills related to teleprocessing. The student shall be provided opportunities to:
    - (A) describe the basic components of a teleprocessing system;
    - (B) demonstrate an understanding of network organization; and
    - (C) understand and identify the use of a teleprocessing system for data exchange in an actual business setting.
  - (2) Concepts and skills related to integrated systems. The student shall be provided opportunities to:
    - (A) understand and identify the advantages, capabilities, and functions of integrated software systems;
    - (B) manipulate data through the use of an integrated system; and
    - (C) use integrated systems to merge data from different applications.
  - (3) Concepts and skills related to graphics. The student shall be provided opportunities to:
    - (A) analyze and interpret business graphs and charts; and
    - (B) demonstrate graphic representation of data.
  - (4) Concepts and skills related to algorithmic approaches to problem solving. The student shall be provided opportunities to:
    - (A) demonstrate the ability to solve problems using appropriate analytical techniques; and
    - (B) demonstrate an awareness of the phases related to systems analysis and design.

- (5) Concepts and skills related to applying logic to programming. The student shall be provided opportunities to:
  - (A) apply the appropriate analytical techniques in solving programming problems;
  - (B) translate the applied solution into a common programming or macro language; and
  - (C) demonstrate the ability to debug and audit a solution.

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**BUSINESS COMPUTER PROGRAMMING I**

(Blended Course)

75.87

(g) **Business Computer Programming I (1, 2, or 3 units) shall include the following essential elements:**

- (1) Concepts and skills related to computer data processing. The student shall be provided opportunities to:
  - (A) demonstrate comprehension of computer-related terminology; and
  - (B) identify the major components of computer systems.
- (2) Concepts and skills related to hardware utilization. The student shall be provided opportunities to:
  - (A) review computer hardware operations;
  - (B) understand categorization of hardware according to size, application, and configurations;
  - (C) reinforce the ability to use basic operating systems; and
  - (D) review the proper care of hardware and software.
- (3) Concepts and skills associated with algorithmic approaches to problem solving. The student shall be provided opportunities to:
  - (A) demonstrate the skills related to structured approaches to problem solving; and
  - (B) translate problems into logical steps using appropriate program design techniques.
- (4) Concepts and skills associated with commands and statements of a programming language. The student shall be provided opportunities to:
  - (A) identify and use variables, constants, and literals;
  - (B) demonstrate the use of input and output statements;
  - (C) demonstrate and use conditional and unconditional branching statements;

- (D) demonstrate and use of looping techniques; and
  - (E) develop and use subroutines for modular programming.
- (5) Concepts and skills related to program execution, testing, debugging, and documentation. The student shall be provided opportunities to:
- (A) demonstrate the ability to execute computer programs with sample data;
  - (B) distinguish between syntax and logic errors;
  - (C) apply standard debugging procedures; and
  - (D) develop internal and external program documentation including record layout and report spacing charts.
- (6) Concepts and skills associated with data storage. The student shall be provided opportunities to:
- (A) demonstrate an understanding of the types of media for data storage;
  - (B) identify methods of storing data on storage devices; and
  - (C) create and maintain a program to store and retrieve data from a file.
- (7) Concepts and skills associated with the social implications of computers. The student shall be provided opportunities to:
- (A) examine issues concerning responsible use of computer systems; and
  - (B) understand the social ramifications of computer applications related to privacy, values, and reliability of systems.

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**BUSINESS COMPUTER PROGRAMMING II**

75.87

- (h) **Business Computer Programming II (1, 2, or 3 units) shall include the following essential elements:**
- (1) Concepts and skills related to application system development. The student shall be provided opportunities to:
    - (A) demonstrate a comprehension of the system development;
    - (B) demonstrate a comprehension of the implementation process; and
    - (D) develop user documentation.
  - (2) Concepts and skills related to structured programming application. The student shall be provided opportunities to:
    - (A) develop an understanding of structured programming;
    - (B) identify the concepts of random and sequential files;
    - (C) translate solutions into major structural elements of a program;
    - (D) translate solutions into language statements for input, processing, and output; and
    - (E) develop ability to use an advanced operating system computer.
  - (3) Concepts and skills related to program testing. The student shall be provided opportunities to:
    - (A) develop test data and testing techniques;
    - (B) apply test data to programs run on the computer; and
    - (C) demonstrate the ability to debug and audit programs.
  - (4) Concepts and skills associated with software implementation and maintenance. The student shall be provided opportunities to:
    - (A) write documentation for maintaining a system;
    - (B) understand the process of implementing a system into production status; and
    - (C) develop an understanding of the programmer responsibilities in program maintenance.

## OFFICE EDUCATION ESSENTIAL ELEMENTS

### OFFICE SUPPORT SYSTEMS

75.87

- (i) **Office Support Systems (1/2 or 1 unit) shall include the following essential elements:**
- (1) Concepts and skills involved in keyboarding. The student shall be provided opportunities to:
    - (A) demonstrate the skill of keyboarding letters, memos, forms, and reports;
    - (B) apply appropriate procedures for proofreading and correcting errors; and
    - (C) demonstrate and identify information processing concepts in keyboarding.
  - (2) Concepts and skills necessary for filing. The student shall be provided opportunities to:
    - (A) demonstrate procedures used in alphabetic, numeric, subject, and geographic filing; and
    - (B) identify procedures used in electronic filing.
  - (3) Concepts and skills necessary for financial activities. The student shall be provided opportunities to:
    - (A) demonstrate the ability to maintain a checking account; and
    - (B) demonstrate the procedures involved in preparing a payroll and preparing tax reports.
  - (4) Concepts and skills necessary for processing mail. The student shall be provided opportunities to:
    - (A) identify the procedures involved in processing incoming and outgoing mail; and
    - (B) identify the procedures involved in transmitting mail electronically.
  - (5) Concepts and skills necessary for effective communications. The student shall be provided opportunities to:
    - (A) identify and use proper business techniques when using the telephone;
    - (B) prepare effective written communications; and
    - (C) identify and use proper language usage skills in business.



- (6) Concepts and skills necessary for effective understanding of business operations. The student shall be provided opportunities to:
  - (A) describe the role of business in the enterprise system; and
  - (B) identify the general responsibilities of employees.
- (7) Concepts and skills necessary for securing and maintaining employment. The student shall be provided opportunities to:
  - (A) demonstrate the skills necessary for job application and interview; and
  - (B) apply human relations skills and the ability to work with others.

## OFFICE EDUCATION ESSENTIAL ELEMENTS

### MICROCOMPUTER APPLICATIONS

75.87

- (j) **Microcomputer Applications (1/2 or 1 unit) shall include the following essential elements:**
- (1) **Fundamental concepts and operational skills related to microcomputer systems. The student shall be provided opportunities to:**
    - (A) understand the background and uses for computer systems;
    - (B) demonstrate competency in touch keyboarding, including technique, speed, and accuracy;
    - (C) understand the components of computer systems, including hardware, operating environments, and application software; and
    - (D) demonstrate competency in microcomputer systems operation.
  - (2) **Concepts and skills necessary for microcomputer-based document production, formatting, and proofreading, at least including use of word processing software and automated spellchecking software. The student shall be provided opportunities to:**
    - (A) identify and use the terminology related to word processing;
    - (B) demonstrate the operational procedures used in document production and page layout; and
    - (C) employ appropriate steps in producing documents.
  - (3) **Concepts and skills necessary for use of microcomputer-based storage, retrieval, manipulation, and reporting information, at least including use of database management software. The student shall be provided opportunities to:**
    - (A) identify and use the terminology related to information management;
    - (B) demonstrate the operational procedures and techniques used in information management software;
    - (C) use microcomputer software to design databases, enter and manipulate information, and produce reports (including graphic display of summary reports); and
    - (D) conduct keyword database searches using logic operations such as AND, OR, and NOT.

- (4) Concepts and skills necessary for use of microcomputer-based computation, analysis, planning, and reporting with numeric data, at least including use of electronic spreadsheets. The student shall be provided opportunities to:
  - (A) identify and use the terminology related to electronic manipulation of numeric data;
  - (B) demonstrate operational procedures and techniques used with microcomputer software for processing numeric data;
  - (C) use microcomputer software for computation, analysis, planning, and reporting (including graphic display) with numeric data; and
  - (D) apply electronic spreadsheet software both through use of prepared templates and with student-designed problem solutions.
- (5) Concepts and skills related to emerging microcomputer-based information processing tools. The student shall be provided opportunities to:
  - (A) understand the concept of exchanging information via telecommunication software, at least including electronic mail, remote bulletin board access, and access of on-line information services;
  - (B) understand and use features of integrated software packages to share data among applications;
  - (C) understand concepts and identify characteristics and uses associated with tools for desktop publishing and generation of presentation graphics;
  - (D) understand concepts related to transporting of textual and nontextual data both locally and remotely, at least including transporting of text and facsimile images electronically, and physical transfer of ultra-high-density storage media; and
  - (E) understand concepts related to networked and multi-user configurations.
- (6) Concepts associated with the social implications of computers. The student shall be provided opportunities to:
  - (A) examine issues concerning responsible use of computer systems;
  - (B) understand the social ramifications of computer applications related to privacy, values, and reliability of systems; and
  - (C) understand the impact of computing on leisure and household management.

**OFFICE EDUCATION ESSENTIAL ELEMENTS**  
**BUSINESS FINANCIAL APPLICATIONS**

75.87

- (k) **Business Financial Applications (1/2 or 1 unit) shall include the following essential elements:**
- (1) Concepts and skills related to banking operations. The student shall be provided opportunities to:
    - (A) identify and understand checking accounts;
    - (B) understand fundamentals of teller operations; and
    - (C) identify and understand bookkeeping systems.
  - (2) Concepts and skills related to credit administration. The student shall be provided opportunities to:
    - (A) understand methods of credit investigation; and
    - (B) understand concepts related to the use of credit analysis.
  - (3) Concepts and skills related to loans. The student shall be provided opportunities to:
    - (A) identify types of loans and purposes; and
    - (B) demonstrate and understand calculation of interest and payoff.
  - (4) Concepts and skills related to legal aspects of banking. The student shall be provided opportunities to:
    - (A) understand customer and consumer services; and
    - (B) understand regulations of financial intermediaries.
  - (5) Concepts and skills related to other financial institutions. The student shall be provided opportunities to:
    - (A) understand and explore savings and loan operations; and
    - (B) understand and explore credit union operations.

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**MEDICAL ADMINISTRATIVE SYSTEMS**

*(Specialty Course--State Board Approval Pending Fall 1990)*

75.87

*(1.1) Medical Administrative Systems (1, 2, or 3 units) shall include the following essential elements:*

- (1) Concepts and skills related to effective communication and production of medical documents. The students shall be provided opportunities to:*
  - (A) identify current terminology relating to the medical industry and to technology;*
  - (B) demonstrate proficiency in business English, spelling, and proofreading;*
  - (C) demonstrate keyboarding speed and accuracy necessary for entry-level employment;*
  - (D) identify and apply correct format for medical correspondence and documents;*
  - (E) apply word processing skills using computer-based tools by employing appropriate steps for document production;*
  - (F) compose and prepare effective medical communications using the document production cycle;*
  - (G) demonstrate proficiency in transcription;*
  - (H) apply proper techniques in business telephone usage;*
  - (I) use proper oral and written language skills relating to business;*
  - (J) recognize classes of mail, special mailing services, and procedures of handling incoming and outgoing mail;*
  - (K) understand concepts related to transporting images electronically; and*
  - (L) identify reprographic copying techniques.*
- (2) Concepts and skills related to information management and control of medical documents. The student shall be provided opportunities to:*
  - (A) identify procedures for controlling manual records by applying terminology and procedures for indexing, coding, and filing alphabetically; and*
  - (B) identify, describe, and/or demonstrate procedures for controlling manual records and computerized information.*

- (3) *Concepts and skills necessary for use of microcomputer-based medical document production, formatting, and proofreading, using word processing software and automated spellchecking, thesaurus, and grammar checking software. The student shall be provided opportunities to:*
  - (A) *identify and use terminology related to word processing;*
  - (B) *demonstrate the operational procedures used in medical document production and page layout; and*
  - (C) *employ appropriate steps in producing medical documents.*
- (4) *Concepts and skills necessary for the operation of office equipment. The student shall be provided opportunities to:*
  - (A) *use the touch system on the electronic calculator/computer to compute business problems related to a medical office; and*
  - (B) *identify reprographic copying techniques.*
- (5) *Concepts and skills related to administering a medical office. The student shall be provided opportunities to:*
  - (A) *perform duties and responsibilities common to most offices.*
- (6) *Concepts and skills related to processing of medical insurance. The student shall be provided opportunities to:*
  - (A) *identify, describe, and/or demonstrate procedures used in processing government claims; and*
  - (B) *identify, describe, and/or demonstrate procedures used in processing private carrier claims.*
- (7) *Concepts and skills related to the daily accounting cycle. The student shall be provided opportunities to:*
  - (A) *demonstrate procedures used in operating a pegboard system; and*
  - (B) *demonstrate procedures used in a computerized medical office accounting system.*

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**LEGAL ADMINISTRATIVE SYSTEMS**

*(Specialty Course--State Board Approval Pending Fall 1990)*

75.87

*(1.2) Legal Administrative Systems (1, 2, or 3 units) shall include the following essential elements:*

- (1) Concepts and skills related to effective communication and production of legal documents. The student shall be provided opportunities to:*
  - (A) identify current terminology relating to the legal business and to technology;*
  - (B) demonstrate proficiency in business English, spelling, and proofreading;*
  - (C) demonstrate keyboarding speed and accuracy necessary for entry-level employment;*
  - (D) identify and apply correct format for legal correspondence and documents;*
  - (E) apply word processing skills using computer-based tools by employing appropriate steps for document production;*
  - (F) compose and prepare effective legal communications using the document production cycle;*
  - (G) demonstrate proficiency in transcription;*
  - (H) apply proper techniques in business telephone etiquette;*
  - (I) use proper oral and written language skills relating to business;*
  - (J) recognize classes of mail, special mailing services, and procedures of handling incoming and outgoing mail; and*
  - (K) understand concepts related to transporting images electronically.*
- (2) Concepts and skills related to information management and control of legal documents. The student shall be provided opportunities to:*
  - (A) identify procedures for controlling manual records by applying terminology and procedures for indexing, coding, and filing alphabetically; and*
  - (B) identify, describe, and/or demonstrate procedures for controlling manual records and computerized information.*



**Business Computer Programming I**  
**APPENDIX C: Office Education Essential Elements**

- (3) *Concepts and skills necessary for use of microcomputer-based legal document production, formatting, and proofreading, using word processing software and automated spellchecking, thesaurus, and grammar-checking software. The student shall be provided opportunities to:*
- (A) *identify and use terminology related to word processing;*
  - (B) *demonstrate the operations procedures used in legal document production and page layout; and*
  - (C) *employ appropriate steps in producing legal documents.*



OFFICE EDUCATION ESSENTIAL ELEMENTS

**INSURANCE ADMINISTRATIVE SYSTEMS**

*(Specialty Course--State Board Approval Pending Fall 1990)*

75.87

(1.3) *Insurance Administrative Systems (1, 2, or 3 units) shall include the following essential elements:*

- (1) *Concepts and skills related to general insurance. The student shall be provided opportunities to:*
  - (A) *understand and explain the origins of insurance;*
  - (B) *identify the different types of losses that property owners are faced with and which are covered by insurance policies;*
  - (C) *explain the different risks that are involved which warrant obtaining insurance;*
  - (D) *differentiate between pure risks and speculative risks that may be encountered by the insured;*
  - (E) *identify requirements of an insurable risk and the items which may prevent obtaining insurance;*
  - (F) *explain society's costs/benefits with insurance and the reasons in favor of purchasing insurance;*
  - (G) *interpret insurance regulations as they pertain to policyholders;*
  - (H) *identify the procedures required for obtaining financial solvency;*
  - (I) *explain the activities of agents and the importance of agents to an agency;*
  - (J) *analyze the methods used by agencies in pricing insurance policies;*
  - (K) *assemble and prepare for submission insurance contracts;*
  - (L) *explain how insurance can be justified as being a product;*
  - (M) *identify and explain the different parts contained in an insurance policy;*
  - (N) *compare and contrast the duties applicable to an agent as those of a broker;*
  - (O) *diagram and explain the path of an insurance policy from the point of contact with the prospect to settlement of a claim; and*

- (P) *simulate the myriad of activities performed by the members of an insurance agency.*
- (2) *Concepts and skills related to automobile insurance. The student shall be provided opportunities to:*
  - (A) *understand and apply insurance terminology correctly in written and oral communication;*
  - (B) *explain the role of insurance concepts to society;*
  - (C) *differentiate the sections contained in a personal automobile policy;*
  - (D) *identify the information contained in an insuring agreement;*
  - (E) *prepare quotations for coverages by use of the rating manual and computer;*
  - (F) *explain the different coverages available and the advantages and disadvantages of each;*
  - (G) *illustrate the effect of variables in preparing quotes, i.e., limits, classifications, territories, and symbols; and*
  - (H) *prepare appropriate documents for insurance transactions.*
- (3) *Concepts and skills related to homeowners insurance. The student shall be provided opportunities to:*
  - (A) *explain the concept regarding eligibility for insurance coverage;*
  - (B) *identify the information contained in the insuring agreement;*
  - (C) *identify and understand personal checking accounts;*
  - (D) *identify and understand personal loan application processes;*
  - (E) *identify and understand different financial institutions;*
  - (F) *identify the role and functions of business management;*
  - (G) *understand the lines of authority; and*
  - (H) *identify effective supervisory techniques.*

OFFICE EDUCATION ESSENTIAL ELEMENTS  
**BANKING AND FINANCIAL SYSTEMS**

*(Specialty Course--State Board Approval Pending Fall 1990)*

75.87

(1.4) **Banking and Financial Systems (1, 2, or 3 units) shall include the following essential elements (specialty course):**

- (1) *Concepts and skills related to effective communication and production of documents used in banking. The students shall be provided opportunities to:*
  - (A) *identify current terminology relating to banking industry and to technology;*
  - (B) *demonstrate proficiency in business English, spelling, and proofreading;*
  - (C) *demonstrate keyboarding speed and accuracy necessary for entry-level employment;*
  - (D) *identify and apply correct format for correspondence and documents used in banking and financial institutions;*
  - (E) *apply word processing skills using computer-based tools by employing appropriate steps for document production;*
  - (F) *compose and prepare effective communications using the document production cycle;*
  - (G) *use proper oral and written language skills relating to business;*
  - (H) *apply proper techniques in business telephone; and*
  - (I) *understand concepts related to transporting images electronically.*
- (2) *Concepts and skills related to information management and control of banking and other financial documents. The student shall be provided opportunities to:*
  - (A) *identify procedures for controlling manual records and applying terminology and procedures for indexing, coding, and filing alphabetically; and*
  - (B) *identify, describe, and/or demonstrate procedures for controlling manual records and computerized information.*
- (3) *Concepts and skills necessary for the operation of office equipment. The student shall be provided opportunities to:*
  - (A) *use the touch system on the electronic calculator/computer to compute business problems related to banking and financial institutions; and*

- (B) *identify reprographic copying techniques.*
- (4) *Concepts and skills related to administration duties in the banking industry. The student shall be provided opportunities to:*
  - (A) *perform duties and responsibilities common to most banking institutions.*
- (5) *Concepts and skills related to banking operations. The student shall be provided opportunities to:*
  - (A) *trace the history of banking and money;*
  - (B) *identify and understand checking accounts;*
  - (C) *understand fundamentals of teller operations; and*
  - (D) *identify and understand bookkeeping systems.*
- (6) *Concepts and skills related to credit administration. The student shall be provided opportunities to:*
  - (A) *understand methods of credit investigation; and*
  - (B) *understand concepts related to the use of credit analysis.*
- (7) *Concepts and skills related to loans. The student shall be provided opportunities to:*
  - (A) *identify types of loans and purposes; and*
  - (B) *demonstrate and understand calculation of interest and payoff.*
- (8) *Concepts and skills related to legal aspects of banking. The student shall be provided opportunities to:*
  - (A) *understand customer and consumer services; and*
  - (B) *understand regulations of financial intermediaries.*
- (9) *Concepts and skills related to other financial institutions. The student shall be provided opportunities to:*
  - (A) *understand and explore savings and loan operations; and*
  - (B) *understand and explore credit union operations.*

**OFFICE EDUCATION ESSENTIAL ELEMENTS**  
**BUSINESS OFFICE CLERICAL**

75.87

(m) **Coordinated Vocational Academic Education (CVAE) Business Office Clerical (1-2 units)** shall include the following essential elements:

- (1) Concepts and skills involved in keyboarding. The student shall be provided opportunities to:
  - (A) demonstrate the skills involved in keyboarding;
  - (B) apply appropriate procedures for proofreading and correcting errors; and
  - (C) demonstrate and identify information processing concepts in keyboarding.
- (2) Concepts and skills necessary for filing. The student shall be provided opportunities to:
  - (A) demonstrate procedures used in alphabetic and numeric filing; and
  - (B) identify procedures used in electronic filing.
- (3) Concepts and skills related to the operation of office equipment. The student shall be provided opportunities to:
  - (A) identify the procedures and equipment used in completing layouts for reprographics;
  - (B) identify the procedures used in operating reprographic equipment;
  - (C) demonstrate the ability to use desktop publishing features;
  - (D) demonstrate the ability to solve mathematical problems using the touch method on the electronic calculator/computer keypad; and
  - (E) demonstrate the ability to enter and retrieve data on a microcomputer.
- (4) Concepts and skills necessary for processing mail. The student shall be provided opportunities to:
  - (A) identify the procedures involved in processing incoming and outgoing mail; and
  - (B) identify the procedures involved in transmitting mail electronically.

- (5) Concepts and skills necessary for effective communications. The student shall be provided opportunities to:
  - (A) develop effective listening skills;
  - (B) identify and use proper business techniques when using the telephone;
  - (C) prepare effective written communications; and
  - (D) identify and use proper language usage skills in business.
- (6) Concepts and skills necessary for securing and maintaining employment. The student shall be provided opportunities to:
  - (A) demonstrate the skills necessary for job application and interview; and
  - (B) apply human relations skills and the ability to work with others.
- (7) Concepts and skills associated with the social implications of computers. The student shall be provided opportunities to:
  - (A) examine issues concerning responsible use of computer systems;
  - (B) understand the social ramifications of computer applications related to privacy, values, and reliability of systems.

## SUGGESTED MINIMUM LIST OF EQUIPMENT, FURNITURE, AND SUPPLIES

(for a quality office education program)

### Equipment and furniture:

A computer or a terminal for each student

- Macintosh/IBM compatible 286 minimum
- 640 K minimum storage
- Two disk drives preferably one 5 1/4 and one 3 1/2 inch
- Hard disk 30 meg.
- Enhanced 101 keyboard with keypad
- High resolution graphics capability
- Several expansion slots

A fully equipped teacher's computer station

Monitors

- Color
- High resolution, at least EGA
- Swivel/tilt monitor bases

Printers (A combination of the following printers may be needed --using one printer to each two computers as a rule of thumb.)

- A 24-pin dot matrix printer with near-letter-quality/high-speed modes
- Letter-quality printers
- Laser printer (preferably postscript)

Other computer peripherals and equipment

- Scanner (if you will be doing a lot of graphics work)
- Automatic switch boxes for switching access to each computer/terminal attached
- Mouse for each computer or terminal
- At least two modems and access to dedicated phone line
- Extra cables for attaching various peripherals

Copy machine

New overhead projector with low heat/good cooling system

LCD computer panel for data projection

Pull-down movie screen

File cabinets and storage cupboards



**Business Computer Programming I**  
**APPENDIX D: Suggested Equipment List**

Adjustable student work stations at appropriate height

Adjustable chairs, preferably rolling

**Supplies**

Covers and keyboard covers

Sufficient number of diskettes for backups of each software program for each computer

Computer paper, both laser and regular cut

Ribbons--standard black and colors

Locking diskette cases

Colored diskettes--each class uses a different color

Computer maintenance kit--including screwdrivers of various sizes, Q-tips, alcohol, screen-wipes, minivacuum, air vacuum

728



## STEPS IN LESSON PLAN DEVELOPMENT

Because of the nature of the business computer programming course, individual lesson plans for each day have not been provided. The sequence of units and production skills in this course are taught in a variety of ways because of the need for each program to meet the needs of its local population. The curriculum outlines at the beginning of each unit provide all of the necessary instructional information for the unit and are designed for your use in preparing lesson plans.

This guide was developed with flexibility, allowing the instructor to teach basic skills in a variety of methods. Although the guide is scoped and sequenced, many units can be incorporated into rotation schedules and can be performed independently by students. A sample lesson format is included.

### Group Instructions

Several units of Business Computer Programming I may be taught by group instruction. These units are called battery units. Battery units are units, taught through group instruction, that all students will need regardless of their career objectives. Some of the battery units that could be taught through the battery system are:

- Social Implications
- Computer-Related Concepts
- Computer Hardware
- Computer Software
- Media and Types of Storage
- Data Representation and Numbering Systems

### Independent Instructions

In independent instruction, the instructor needs to schedule students on all equipment or work stations related to the student's interest or career objective. Students' work needs to be individualized at these stations. Some students will be able to complete the project within two or three days, while others may not complete the work within a week. Avoid giving the faster student busy work. Try to provide training that will assist students in becoming more knowledgeable and more productive. Practice sets and software packages are ideal for independent instruction. Units that lend themselves to independent instruction include the following:

- Computer Operation and Maintenance
- Problem Solving and Applications
- Programming Concepts and Applications
- Data Storage Applications

### Integrated Software and Simulations

The automated office is rapidly moving toward the systems concept. In this approach many information processing tools are used in an integrated manner. It is important for students to learn to function in this atmosphere. This can be done by the office simulation strategy, using integrated software packages.

Units that lend themselves to simulations through the use of integrated software include the following:

- Computer-Related Concepts
- Computer Operation and Maintenance
- Problem-Solving Concepts and Applications
- Data Storage Applications

### Evaluation

The State Board of Education Rules for Curriculum states that students should be evaluated on a 0-100 scale with 70 and above receiving course credit. At the beginning of the year, you should establish the grading system for your class. Consideration should be given to how individualized instruction, battery instruction, attendance, etc., will affect the students' grades for the grading period.

In evaluating students on independent instruction, it is impossible to evaluate each job assigned. Therefore, on some assignments, answer sheets may be given to the students so they may check their answers with the teacher's key. On some assignments, the teacher might grade for specifics. For example, grade one program for documentation, one for keying ability, one for logic, one for syntax, etc. Since students may not be completing the same assignments, students may be given a grade sheet and assignment sheet for each rotation.

**SAMPLE LESSON FORMAT**

COURSE \_\_\_\_\_ INSTRUCTOR \_\_\_\_\_

Unit Title \_\_\_\_\_

LESSON TITLE \_\_\_\_\_

Estimated Time \_\_\_\_\_ (Hours or Days)

Essential Element(s) \_\_\_\_\_

Domain(s)      \_\_\_\_\_ Cognitive      \_\_\_\_\_ Psychomotor      \_\_\_\_\_ Affective

Basic Skill(s)      \_\_\_\_\_ Reading      \_\_\_\_\_ Writing  
\_\_\_\_\_ Mathematics

Student Competency(ies) \_\_\_\_\_

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

**Teacher Preparation**

**Resources**

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

### Teacher Presentation

## Teaching Strategies

## Student Activities

## Resources

**Assignment(s)**

---

---

---

---

---

---

---

---

---

---

**Closure**

---

---

---

---

---

**Evaluation**

---

---

---

---

---

---

---

---

## PLANNING THE LESSONS

### Divisions of the Lesson Plan

The best kind of teaching is based on good planning, good presentation, and a well-disciplined class. Good planning is based on a series of steps. Although individual lesson plans vary, the most common is the developmental plan.

1. **Unit Title**--Unit title identifies the broad area of which the specific lesson is a part.
2. **Lesson Title**--The lesson title identifies a category of the unit for which instruction is delivered.
3. **Estimated Time**--The time frame assists the instructor in determining the scope of each lesson so that the students are able to perform the competencies.
4. **Essential Elements**--Essential element or elements might be included within a lesson or a complete lesson. Identifying essential elements, if any, will assist the instructor in planning for instruction of the essential elements.
5. **Domains**--Lesson consists of three kinds: (1) cognitive domain (knowledge or information), (2) psychomotor domain (skills), and (3) affective domain (values or attitudes). Determining the kind of lesson assists the instructor in selecting the necessary teaching materials, strategies, and tools needed for a particular lesson.
6. **Basic Skills**--In preparing each lesson, the instructor needs to identify the basic skills which are a part of the lesson.
7. **Student Competencies**--Since the curriculum is performance-based instruction, students are expected to acquire specific competencies. These competencies need to be identified for each lesson.
8. **Teacher Preparation**--It is here that the instructor decides on the resources: supplies, materials, equipment, and other instructional items which will be needed for the lesson. The preparation might include charts, tables, books, field trips or other kinds of information. Here also, the preparation needed by students should be considered. Perhaps there has been a reading assignment or a notebook requested. Whatever the student needs for the lesson should be determined at this point.



9. **Teacher Presentation**--This is the part of the lesson in which the thing to be learned is presented. Presentation should proceed, step by step, simple to complex, with teaching points interspersed in the proper sequence. The instructor should think out the procedure or explanation beforehand and then develop it. Part of this process is selecting appropriate teaching strategies. Examples are: demonstration, questioning, discussion, drill, and lecture. Strategies should be selected based (1) on the kind of lesson being taught and (2) on the learning needs of the students for whom the instruction is intended.
10. **Student Activities**--Practicing the thing to be learned is the keynote of successful teaching and learning in the field of vocational/technical education. The activities assigned to the students are the part of the lesson that makes use of this principle by providing an opportunity for the students to do something about the thing learned.
11. **Closure**--Comes at the end of the lesson and ties everything together. Closure could be called a summary; however, with closure the instructor guides the students into seeing how the parts contribute to make the whole.
12. **Assignment**--The assignment part of the lesson may be considered a continuation of the student activities. Assignments provide a meaningful further application of the thing just learned. It is an aide to retention. One of the laws of learning is repetition. The assignment may also consist of preparation for the next lesson.
13. **Evaluation**--Both the teacher and the student need some method of feedback for establishing if learning has taken place. This system is broadly referred to as "evaluation."

Student evaluation is usually thought of as:

- 1) "paper and pencil" tests, and
- 2) norm-referenced testing--comparing students on a scale of 0-100.

Student evaluation in vocational-technical subjects should include a variety of methods. Some suggestions are:

**Norm-Referenced Testing.** Norm-referenced testing is a form of educational measurement that ascertains an individual's performance in relation to the performance of other individuals on the same measuring device. Because the individual is compared with some normative group, such measures are described as norm-referenced. Examples: standardized objective tests, and teacher-prepared essay and objective tests.

**Criterion-Referenced Testing.** Criterion-referenced testing is a form of educational measurement that ascertains an individual's status with respect to some criterion or performance standard. Because the individual is compared with some established criterion, rather than with other individuals, these measures are described as "criterion-referenced." Examples:

1. Attached is a simple program. By the next class day, enter and run the program. The program must be debugged. A score of 100% is required.
2. Attached is a list of programming terminology. Define the terms. A score of 75% or higher is required. Maximum time is forty minutes.

Since the curriculum guide is performance-based instruction with specified student competencies, criterion-referenced testing is considered an excellent method for measuring student performance.

There are three essential steps to remember in implementing criterion-referenced measurement. They are:

1. Prior to instruction, prepare a set of instructional objectives for the unit.
2. Select appropriate assessment approaches/techniques to assess students' mastery of the stated objectives. The appropriateness of the technique selected will largely be determined by the nature of the skill or competency specified in the objective.
3. Match particular assessment techniques selected with the performance behaviors delineated in the instructional objectives. It is important that the specific test item corresponds to the level and type of behavior specified in the objective. In other words, test on the same level that you teach. The levels of the three domains are broken out in this Appendix.

**Other methods of evaluation are:** Teacher observation of skill and/or attitude, teacher-assigned projects, production measures, and accuracy measures.



## MAJOR CATEGORIES IN THE COGNITIVE DOMAIN

### **Evaluation**

involves acts of  
decision making,  
judging, or selecting  
based on criteria  
and rationale  
**requires synthesis in  
order to evaluate**

### **Synthesis**

combines elements to  
form new entity from  
original one  
**requires analysis in  
order to synthesize**

### **Analysis**

separates whole into its  
parts, until relationship  
among elements is clear  
**requires ability to  
apply information in  
order to analyze**

### **Application**

uses information in a  
situation different from  
original learning context  
**requires comprehen-  
sion of information  
in order to apply in  
new situation**

### **Comprehension**

interprets, translates,  
summarizes, or para-  
phrases given information  
**requires knowledge in order  
to demonstrate  
comprehension**

### **Knowledge**

recognition and recall  
of facts and specifics

## VERBS FOR STATING BEHAVIORAL OBJECTIVES

### COGNITIVE DOMAIN

1. Knowledge--remembering previously learned materials
 

|          |       |        |           |
|----------|-------|--------|-----------|
| Cite     | Label | Name   | Reproduce |
| Define   | List  | Quote  | Pronounce |
| Identify | Match | Recite | State     |
2. Comprehension--ability to grasp the meaning of material
 

|          |               |           |            |
|----------|---------------|-----------|------------|
| Alter    | Explain       | Portray   | Simplify   |
| Change   | Give examples | Rephrase  | Substitute |
| Convert  | Illustrate    | Represent | Summarize  |
| Depict   | Interpret     | Restate   | Translate  |
| Describe | Paraphrase    | Reword    | Vary       |
3. Application--ability to use learned material in new and concrete situations
 

|             |          |         |         |
|-------------|----------|---------|---------|
| Apply       | Discover | Manage  | Relate  |
| Compute     | Employ   | Predict | Show    |
| Demonstrate | Evidence | Prepare | Solve   |
| Direct      | Manifest | Present | Utilize |
4. Analysis--ability to break down material into its component parts so that its organizational structure can be understood
 

|           |               |             |           |
|-----------|---------------|-------------|-----------|
| Ascertain | Diagnose      | Distinguish | Outline   |
| Analyze   | Diagram       | Divide      | Point out |
| Associate | Differentiate | Examine     | Reduce    |
| Designate | Discriminate  | Find        | Separate  |
| Determine | Dissect       | Infer       |           |
5. Synthesis--ability to put parts together to form a new whole
 

|          |            |           |            |
|----------|------------|-----------|------------|
| Combine  | Devise     | Originate | Revise     |
| Compile  | Expand     | Plan      | Rewrite    |
| Compose  | Extend     | Pose      | Synthesize |
| Conceive | Generalize | Propose   | Theorize   |
| Create   | Integrate  | Project   | Write      |
| Design   | Invent     | Rearrange |            |
| Develop  | Modify     |           |            |
6. Evaluation ability--to judge the value of material for a given purpose
 

|          |           |          |       |
|----------|-----------|----------|-------|
| Appraise | Conclude  | Critique | Judge |
| Assess   | Contrast  | Deduce   | Weigh |
| Compare  | Criticize | Evaluate |       |

## MAJOR CATEGORIES IN THE PSYCHOMOTOR DOMAIN

### **Naturalization**

completes one or more skills with ease and becomes automatic with limited physical or mental exertion

### **Articulation**

combines more than one skill in sequence with harmony and consistency

### **Precision**

reproduces a skill with accuracy, proportion, and exactness; usually performed independent of original source

### **Manipulation**

performs skill according to instruction rather than observation

### **Imitation**

observes skill and attempts to repeat it

## VERBS FOR STATING BEHAVIORAL OBJECTIVES

### PSYCHOMOTOR DOMAIN

1. Perception--becoming aware of the action to be performed through sensory stimulation

|         |           |       |
|---------|-----------|-------|
| Detect  | Note      | See   |
| Examine | Notice    | Sense |
| Feel    | Observe   | Smell |
| Handle  | Perceive  | Taste |
| Hear    | Recognize | Touch |
| Hold    |           |       |

2. Set--becoming ready to act mentally, physically, and emotionally

|             |          |         |
|-------------|----------|---------|
| Adjust      | Locate   | Set     |
| Check       | Place    | Set up  |
| Confirm     | Position | Sit     |
| Decide      | Ready    | Situate |
| Estimate    | Regulate | Stand   |
| Familiarize | Select   | Station |

3. Guided response--performing the action under supervision through imitation or trial and error

|            |          |           |
|------------|----------|-----------|
| Attempt    | Follow   | Repeat    |
| Carry out  | Imitate  | Reproduce |
| Copy       | Mimic    | Test      |
| Duplicate  | Practice | Try       |
| Experiment |          |           |

4. Mechanism--performing the act habitually with some degree of confidence

|           |            |          |
|-----------|------------|----------|
| Acquire   | Gain       | Pace     |
| Assemble  | Improve    | Perform  |
| Complete  | Maintain   | Proceed  |
| Conduct   | Make       | Produce  |
| Construct | Manipulate | Progress |
| Do        | Measure    | Use      |
| Execute   | Operate    |          |

5. Complex overt response--performing the act automatically with high level of skill

|            |         |           |
|------------|---------|-----------|
| Achieve    | Exceed  | Refine    |
| Accomplish | Excel   | Succeed   |
| Advance    | Master  | Surpass   |
| Attain     | Perfect | Transcend |
| Automatize | Reach   |           |

## MAJOR CATEGORIES IN THE AFFECTIVE DOMAIN

**Characterizing**  
total behavior is  
consistent with  
values internalized

**Organizing**  
committed to set of  
values as displayed  
by behavior

**Valuing**  
displays behavior  
consistent with  
single belief or  
attitude in situa-  
tions where he/she is  
not forced to comply  
or obey

**Responding**  
complies with given  
expectations by  
attending or  
reacting to stimuli  
or phenomena; i.e.,  
interests

**Receiving**  
aware of; passively  
attending to certain  
phenomena and  
stimuli; i.e.,  
listening

## VERBS FOR STATING BEHAVIORAL OBJECTIVES

### AFFECTIVE DOMAIN

1. Receiving--attending and becoming aware of a situation, idea, or process through the sensory organs
 

|             |                |          |
|-------------|----------------|----------|
| Acknowledge | Look           | Tolerate |
| Attend      | Pay attention  | View     |
| Be aware    | Receive        | Watch    |
| Listen      | Show alertness |          |
  
2. Responding--doing something about a situation, idea, or process
 

|             |             |           |
|-------------|-------------|-----------|
| Agree (to)  | Cooperate   | React     |
| Answer      | Discuss     | Read      |
| Ask         | Follow up   | Reply     |
| Assist      | Indicate    | Request   |
| Be willing  | Inquire     | Respond   |
| Communicate | Obey        | Seek      |
| Comply      | Participate | Visit     |
| Consent     | Pursue      | Volunteer |
| Contribute  | Question    |           |
  
3. Valuing--developing attitudes by choosing, prizing, cherishing
 

|         |         |          |
|---------|---------|----------|
| Accept  | Desire  | Form     |
| Adopt   | Display | Initiate |
| Approve | Endorse | Prefer   |
| Choose  | Exhibit | Sanction |
| Commit  | Express |          |
  
4. Organization--arranging values in priority according to a system
 

|            |           |           |
|------------|-----------|-----------|
| Adapt      | Establish | Organize  |
| Arrange    | Formulate | Rank      |
| Categorize | Group     | Rate      |
| Classify   | Order     | Systemize |
  
5. Characterization--choosing the same values consistently to form a philosophy of life
 

|               |             |           |
|---------------|-------------|-----------|
| Act           | Endure      | Preserve  |
| Advocate      | Exemplify   | Recommend |
| Be consistent | Function    | Remain    |
| Behave        | Habituate   | Retain    |
| Characterize  | Incorporate | Reveal    |
| Conform       | Influence   | Serve     |
| Continue      | Justify     | Support   |
| Defend        | Maintain    | Uphold    |
| Devote        | Pattern     | Urge      |
| Disclose      | Persist     | Verify    |
| Encourage     |             |           |

## COURSE COMPETENCY CHECKLIST

(Mastery = 4 and Nonmastery = 1)

### COMPETENCIES

#### *I. Introduction to Business Computer Programming*

##### *1. Introduction To Course* NONE

##### *2. Social Implications*

- 4 3 2 1 1. Identify terminology related to computer crime, computer users, and new computer devices and techniques.
- 4 3 2 1 2. Identify the major issues and technological developments related to computers and society.
- 4 3 2 1 3. Describe the ethical and legal issues that have arisen as a result of the computer revolution.
- 4 3 2 1 4. Identify privacy issues involved with electronic data storage.
- 4 3 2 1 5. Explain the reasons for concern about the reliability of computer systems.

##### *3. Computer-Related Concepts*

- 4 3 2 1 1. Identify basic computer-related terminology.
- 4 3 2 1 2. Outline the functions of data processing.
- 4 3 2 1 3. Illustrate the parts of the information processing cycle, and explain how each part is interdependent within the cycle.
- 4 3 2 1 4. Illustrate parts of the computer processor.
- 4 3 2 1 5. Compare the similarities and differences among various computer systems.
- 4 3 2 1 6. Outline the events in history pertinent to breakthroughs in data processing.
- 4 3 2 1 7. Describe and categorize characteristics of computer languages.

#### *II. Computer Hardware Utilization*

##### *1. Computer Hardware*

- 4 3 2 1 1. Identify the types of computers.
- 4 3 2 1 2. Describe primary memory.
- 4 3 2 1 3. Describe various input devices.
- 4 3 2 1 4. Identify various output devices.
- 4 3 2 1 5. Identify other hardware devices.

##### *2. Computer Software*

- 4 3 2 1 1. Describe the characteristics and functions of an operating system.
- 4 3 2 1 2. Identify and categorize applications that would utilize batch processing and real-time processing.
- 4 3 2 1 3. Identify a compiler, interpreter utilities, and virtual storage.
- 4 3 2 1 4. Discuss the different versions of the microcomputer operating systems.
- 4 3 2 1 5. Identify the major components of the DOS/VS operating system.

### COMPETENCIES

#### *3. Computer Operation and Maintenance*

- 4 3 2 1 1. Identify terminology related to computer operation and maintenance.
- 4 3 2 1 2. Operate and maintain machines.
- 4 3 2 1 3. Monitor and report environmental problems.
- 4 3 2 1 4. Inventory resources and report findings.
- 4 3 2 1 5. Handle media and support tools.
- 4 3 2 1 6. Monitor hardware performance.
- 4 3 2 1 7. Schedule workflow.
- 4 3 2 1 8. Maintain data and physical security.

#### *III. Program Development*

##### *1. Problem-Solving Concepts and Applications*

- 4 3 2 1 1. Define structured programming and related terminology.
- 4 3 2 1 2. Identify three types of control structures.
- 4 3 2 1 3. Demonstrate problem-solving techniques.
- 4 3 2 1 4. Utilize flowcharting and top-down analysis.

##### *2. Programming Concepts and Applications*

###### *MODULE A: SETUP PROCEDURES*

- 4 3 2 1 1. Demonstrate the ability to power-up, load the compiler, and power-down the computer.
- 4 3 2 1 2. List the steps of problem solving.

###### *MODULE B: INTRODUCTION TO PROGRAMMING*

- 4 3 2 1 1. Identify keywords and commands.
- 4 3 2 1 2. Code, key, save, load, and execute a program.
- 4 3 2 1 3. Edit program lines.
- 4 3 2 1 4. Write simple programs using the keywords REM, PRINT, END.

###### *MODULE C: DOCUMENTATION, DEBUGGING, AND TESTING*

- 4 3 2 1 1. Explain the benefits of documentation.
- 4 3 2 1 2. Identify the three major types of errors.
- 4 3 2 1 3. Develop test data for a program.

###### *MODULE D: USING ARITHMETIC*

- 4 3 2 1 1. Define the types of variables.
- 4 3 2 1 2. Use arithmetic operators.
- 4 3 2 1 3. Develop arithmetic equations.
- 4 3 2 1 4. Write programs using the keyword LET.

###### *MODULE E: INPUT CONCEPTS*

- 4 3 2 1 1. Demonstrate the use of data terminators.

**Business Computer Programming I**  
**APPENDIX F: Course Competency Checklist**

**COMPETENCIES**

- 4 3 2 1 2. Write programs using the keywords READ/DATA, IF...THEN, GOTO, and INPUT.  
 4 3 2 1 3. Demonstrate an output report by using proper headings (major, minor, and column), and PRINT USING.

**MODULE F: BRANCHING**

- 4 3 2 1 1. Identify the relational operators.  
 4 3 2 1 2. Identify the logical operators.  
 4 3 2 1 3. Describe the difference between a conditional branch and an unconditional branch.  
 4 3 2 1 4. Write programs using the conditional branch.

**MODULE G: TOTALS**

- 4 3 2 1 1. Define totaling and counting.  
 4 3 2 1 2. Demonstrate how subtotals are accumulated and printed.  
 4 3 2 1 3. Write programs to demonstrate totaling, counting, and subtotals.

**MODULE H: MODULE PROGRAMMING**

- 4 3 2 1 1. Define modular programming.  
 4 3 2 1 2. Write programs to use a main module and submodules.

**MODULE I: CONTROLLED LOOPS**

- 4 3 2 1 1. Write programs using the keywords FOR/NEXT.

**COMPETENCIES**

**MODULE J: ADVANCED CONCEPTS**

- 4 3 2 1 1. Write programs using tables to store data.  
 4 3 2 1 2. Write programs using sort routines.  
 4 3 2 1 3. Write programs using advanced character handling.  
 4 3 2 1 4. Write programs using graphics.

**IV. Data Storage**

**1. Media and Types of Storage**

- 4 3 2 1 1. Define primary and secondary storage.  
 4 3 2 1 2. Identify secondary storage devices and media.  
 4 3 2 1 3. Distinguish between volatile and nonvolatile memory.  
 4 3 2 1 4. Differentiate between file organizations and file access methods.

**2. Data Storage Applications**

- 4 3 2 1 1. Identify data elements--field, record, and file.  
 4 3 2 1 2. Code a program to create a sequential and/or random data file.  
 4 3 2 1 3. Code a program to retrieve data and produce a report from a data file they created.

**3. Data Representation and Numbering Systems**  
 NONE



## ACKNOWLEDGMENT

The module 7 Overview of the Vocational Special Needs Teachers Training Curriculum has been reprinted with permission as this appendix. Teachers are encouraged to refer to the original publication to learn more about working with special needs populations. The character font size and line spacing have been changed in this appendix to save space.

The project was developed in 1987 by Craig R. Colby, Ed.D., of Texas A&M University. Linda H. Parrish, Ph.D., Texas A&M University, and Jerry L. Wircenski, Ph.D., North Texas State University, were the project directors.

The vocational special needs teacher training curriculum was developed pursuant to a contract with the Texas Education Agency's Vocational Education Special Projects. Contractors or grantees undertaking such projects under Agency sponsorship are encouraged to freely express their professional judgment in the conduct of the project. Points of view or opinions stated do not, therefore, necessarily represent official Texas Education Agency position or policy.

This information is presented as an appendix to help office education teachers with planning the modification of instruction for special population. Many of the concepts and ideas presented are sound practices to be used with normal populations.

The entire 10-module training curriculum is available through the Extension, Instruction and Materials Center, University of Texas at Austin, P.O. Box 7218, Austin, Texas 78713-7218.

Some of the text has been modified or omitted to make it more relative to office education.

## CURRICULUM MODIFICATION

The term *vocational curriculum* can be defined as the organized content of a particular discipline which a student must master to attain an occupational goal. It involves the sum total of all experiences and learning activities encountered in the classroom. A curriculum is concerned with results. It encompasses "what is to be taught" and "what is to be learned" (Stern-Otazo, 1980; Goh & Sitlington, 1983).

Each student has a mental capacity, physical skills, and various needs which vary from one individual to the other. In addition, students exhibit different learning styles. All good teachers recognize this and will work toward developing a classroom environment which is characterized by a deep respect for the learner, acceptance of student, fulfillment of individual needs, and individual development. This is what is meant by the teacher who has become "learner-centered" and not "content-centered" (Pine & Boy, 1977). Jones (1977) eloquently describes recognizing individual worth and individual development as being the basis for curriculum development as follows:

**Individual development** as a basis for curriculum perspective would lead us to teach discovery of principles and sacredness of the individual personality. It would teach the learners to view themselves as self-actualizing individuals in a setting designed to support and encourage personal development to the ultimate of one's potential. Attempts to improve the local community would center around providing a richer and fuller life for personal growth and development. Good people make good communities, good states, good nations, and a good world. By emphasizing development of individuals, such a curriculum would hope to plant the seed for a better world in the future.

Therefore, all learners should be perceived as very "special" individuals who one day will take their place in society. It is the responsibility of educators to assist all learners (in spite of their shortcomings) to become productive members of society. This can be accomplished by helping each student establish attainable career goals and then providing meaningful experiences through which the student can achieve these goals. Every student should be treated with dignity, as an individual who has unique strengths and interests in life. It is not necessary or even realistic to expect every student to do and know all the same things. As stated by Jones (1977), "Learners should become worthwhile persons in their own right,...it is not necessary that they all become miniature images of the same pattern."

### Purpose of Curriculum Modification

When working with students with special needs, a learner's curriculum may need to be modified or adjusted to take into account major differences which could be found among these individuals. The term *curriculum modification* can be defined as tailoring an existing

curriculum to the unique needs of students as they pursue an occupational goal (Sarkees & Scott, 1985; Goh & Sitlington, 1983). However, it should be strongly emphasized that modifying a curriculum does not mean that an educator should lower or alter the standards of his or her program. What it does mean is that the Admission Review and Dismissal (ARD) Committee (for handicapped students) or coordinated planning team (for disadvantaged and LEP students) determines which essential elements in the regular school curriculum would be appropriate for a student's needs. The *State Board of Education Rules for Curriculum* offers the following key points (Texas Education Agency, 1985):

1. For students with special needs, school districts shall provide modifications in methodologies, pacing, and materials to ensure that instruction is delivered, as appropriate, in the essential elements.
2. Students with special needs shall be instructed in those same essential elements in a manner appropriate to their needs.
3. School districts shall implement the provisions (under Chapter 75) in a manner which will enable all students to participate actively in a well-balanced curriculum and master, to the best of their abilities, the knowledge, skills, and competencies established through the essential elements.
4. Specialized personnel and regular program personnel shall be jointly responsible for the cooperative delivery of effective instruction in the essential elements, as appropriate.

It is important to re-emphasize that special needs students, as all students, have capabilities which widely vary. The challenge parents, educators, and vocational counselors face is to help each learner reach occupational goals which are within the student's potential.

### **The Curriculum Modification Process**

For the purpose of this overview, the competency-based curriculum modification model has been selected for discussion.

For information regarding the steps involved in the modification process including the establishment of an ARD committee and developing the IEP plans, refer to the 1989 TEA publication, *Office Education Handbook*.

## The Competency-Based Curriculum

Over the years, the concept of competency-based curriculum has increased in popularity largely due to the emphasis on **accountability** in education (Choate, et al., 1987). In addition, since the passage of P.L. 94-142, vocational educators have had to adjust their curricula to take into account the capabilities of special needs learners.

The underlying principle behind the competency-based curriculum is to assist the student in becoming a more effective person. This entails helping the student develop skills or competencies which will promote employment opportunities and a higher self-esteem within the student (Brolin, 1976; Lippmann & Porter, 1976). The ability to succeed is what makes competency-based learning so valuable for students with special needs (Kok, 1978). With competency-based learning a student does not have to wait until the end of the program to experience success. The sequential steps called for throughout the program afford the possibility of success on a day-to-day basis (Lippmann & Porter, 1976).

## Responsibilities of the Vocational Educator

The primary responsibility of the vocational educator in the curriculum modification process is to help the special needs student establish and reach realistic occupational goals. In an effort to accomplish this, the vocational educator must identify the following (Sarkees & Scott, 1985):

1. Pertinent information about what will be taught in a vocational program.
2. Necessary requirements for program completion.
3. Multiple exit points (jobs that the learner can prepare for in a vocational program).
4. Entry level competencies expected by employers for each of the exit points.

It is also the responsibility of every vocational educator to establish an effective and wholesome learning environment for their students. Providing this environment may involve modifying the vocational facility, equipment, tools and/or instruction. This information will be provided in the following section.

## MODIFICATION OF THE LEARNING ENVIRONMENT

The term *learning-environment* refers to both the **physical** and **instructional** setting of the vocational classroom/laboratory (Bell, 1984). It is the responsibility of each vocational educator to work at developing a learning environment which will contribute to the overall well-being and success of every student. The key factors found in a wholesome learning environment are those which promote:

1. **Safety, organization, and physical comfort.** In the vocational laboratory, it is absolutely crucial that each student abide by safety rules set by the instructor. In addition, the laboratory should be kept well organized and clean. A laboratory which is safe, well organized, and physically comfortable helps promote high morale and good work habits.
2. **Respect and acceptance.** Within an effective learning environment, students will show respect not only for the instructor but for each other as well. The instructor should be the student's best role model. The care and acceptance given by the instructor to each student (disabled or nondisabled) will go a long way in promoting peer acceptance, a positive attitude, and healthy self-esteem.
3. **Discipline.** A wholesome learning environment is one which promotes discipline and a positive self-concept within each student. Inappropriate behavior has the potential to seriously affect classroom safety, morale, and the overall learning process. Ways in which the instructor can help the student modify inappropriate behavior include: (a) utilizing the time-out technique, (b) developing student-teacher contracts, (c) ignoring negative behavior--whenever possible, and (d) rewarding appropriate behavior with approval and praise.
4. **Flexibility.** An effective learning environment is flexible and can adapt to a special student's unique needs (i.e., slower rate of learning, inability to access regular tools/equipment, or remedial needs in reading or math for instance). A flexible environment fosters individualized programs through which a special learner can develop marketable skills for employment.

### Modifying Facilities, Equipment, and Tools

In the past, many special needs learners have been denied their basic right to a public education because of architectural barriers, nonadaptive facilities, equipment, and tools (Phelps & Lutz, 1977). However, federal legislation has done a great deal to provide equal access to educational programs for every learner. For example, according to

Section 503 of the Rehabilitation Act of 1973, each federally funded program or activity in which a handicapped individual can enroll and profit must be made accessible to the student (Sarkees & Scott, 1985). In addition, the Carl Perkins Act of 1984 has emphasized the importance of providing supplemental services for special needs learners, which include the adaptation of facilities, equipment, curriculum, and instruction (Cobb & Kingbury, 1985).



## MODIFICATION OF INSTRUCTION

The primary purpose of this section is to provide the vocational educator with teaching strategies and techniques that will be useful when instructing students with special needs. It is important to remember that the methods and approaches used by educators when instructing non-special needs students may not be very effective with students with special needs (Gleiss-Monar & Sitlington, 1983). Therefore, individualizing instruction becomes vitally important in the successful instruction of special learners.

Within this section, the following topics will be discussed: (a) learning styles, (b) teaching strategies, (c) remediation techniques, and (d) Brown's 3-Cs Instructional Mode.

### Learning Styles

Each special learner has unique needs, and these needs affect how the student can best learn. For example, if a student is blind or has difficulty with "visual" learning, the instructor should stress the learner's sense of sight. Therefore, the manner in which a student with special needs learns ceases to be a preference, but becomes instead a **necessity** (Kok, Jernigan, & Hull, 1978). Further, one of the main philosophies behind individual learning styles is that learning should be a pleasant and successful experience. Educators, therefore, should adjust their style of teaching to allow students the benefit of learning in their preferred manner (Kok, Jernigan, & Hull, 1978).

### Identifying a Student's Learning Style

The following are methods in which an educator can identify a student's learning style:

1. Informal observation. One of the simplest and possibly best ways to identify a student's learning style is through informal observation. By observing the student as he or she negotiates classroom/laboratory learning activities, the educator can collect valuable information with regard to the student's learning style. For example, the educator may observe that a disadvantaged student answers questions in class and does well with hands-on activities but repeatedly fails written exams. If the primary intent of the instruction is to reach vocational skills rather than reading and writing skills, the instructors could modify their style of teaching by allowing the student to take oral and/or hands-on exams (Kok, Jernigan, & Hull, 1978; Sarkees & Scott, 1985).
2. Teacher consultation. Consulting other teachers (i.e., special education teachers, resource room teachers, support personnel, paraprofessionals, and other voca-

tional education teachers) can be an excellent source for information regarding a student's learning style (Sarkees & Scott, 1985).

3. Student/parent interviews. Interviewing the student and student's parents/guardians can result in valuable insights as to the student's best learning modes (Sarkees & Scott, 1985).
4. Standardized inventories. Standardized inventories provide an alternative for students who are shy or uncomfortable during a personal interview. Information received from interest, personality, and learning style inventories can provide valuable information with regard to the student's learning style (Kok, Jernigan, & Hull, 1978; Sarkees & Scott, 1985).

### DETERMINING LEARNING STYLES

For many years educators have recognized the need for alternative instructional methods and techniques to meet the needs of the variety of students in our heterogeneously grouped classrooms. We have become familiar with a variety of methods and techniques, but, in many cases, their misapplication has proven unproductive. Recognizing and defining the style in which a person learns is as important to the learning process as the diagnostic tests required by a medical doctor are to the healing process. Many teachers share the frustration of knowing that we are failing to meet the needs of some of the students in our classrooms. One reason could be the lack of understanding and accommodating the unique learning styles of our students--particularly those we are not reaching.

The CITE Learning Styles Instrument was formulated by teachers for the use of teachers in determining the styles preferred by their students. The Learning Styles Instrument is divided into three main areas: information gathering/receiving, social work conditions, and expressiveness preference. The Instrument is simple to administer, score, interpret and utilize in accommodating individual differences. It also is an instrument that offers immediate data, as well as accompanying information with suggestions for appropriate implementation. There are provisions being made for scoring, charting, and interpreting by the computer.

If we are to allow for individual differences, we must consider ways of learning, working, and reporting as three (3) important areas. In the area of learning, the CITE Learning Styles Instrument investigates five (5) specific subtopics. They are:

1. Auditory Language: the way a student hears words; processing spoken words.



2. Visual Language: the way a student sees words; processing written language.
3. Auditory Numerical: the way a student hears numbers; processing spoken numerical values.
4. Visual Numerical: the way a student sees numbers; processing written numerical values.
5. Auditory-Visual-Kinesthetic: the way a student learns by doing or involvement. Emphasizing the 'experiencing' or manipulative learning style which is almost always accompanied by either auditory stimuli, visual stimuli, or a combination of both.

The area of working considers whether a student likes to work, or learn, in a group or alone. They are appropriately titled as follows:

1. Group learner: A student who likes to work with at least one other person when there is important work to be done.
2. Individual learner: A student who works and thinks best alone. This student is usually a self-starter and frequently finds working with other students distracting.

The area of individual expressiveness considers how a student prefers to express himself. Basically, they fall into one of these two broad categories:

1. Oral expressive: A student who prefers to say what he knows. Usually, answers or explanations are better given orally; however, some students may indicate this preference simply because they are too lazy to write things down.
2. Written expressive: A student who prefers to write down answers or explanations. Students who exhibit a reflective cognitive learning style may prefer this method.

The administration of the CITE Learning Styles Instrument is simple and can be done in a classroom with 30 or 35 students in less than 20 minutes. Students need only to have a pencil and a copy of the CITE Learning Styles Instrument. Directions should make very clear that the statements are not questions with right or wrong answers. They are just statements, and the responses should represent the way the student feels. Emphasize that no grades are given, and, if they are to score it themselves, they will do the actual scoring and charting results. Explain how they are to mark the answers and do the Learning

Styles example from the front page of the Instrument. Emphasize that they must make only one choice for each statement, but that they must respond to every statement. Usually their first choice is the best. Explain that as they read each statement, you will read it aloud. This is to accommodate those who prefer auditory to visual stimuli or for nonreaders, and allow approximately 12 seconds to read and respond to each statement.

Encourage and allow questions from the students if they do not understand. Special solicitations for questions should be extended before the Learning Styles example is done, after it is done, before the first statement from the Instrument is read, after the third statement is read, after the last statement is read, and any other time when the need is apparent. After the Instrument is completed, it is necessary to check for one response to every statement. The students should be asked to put their names on each page of the Instrument, and, if the students are to score their own, the scoring procedure will begin.

The process of scoring the CITE Learning Styles Instrument using the computer is still being worked out. If you are a teacher in the Unified School District 259, computer scoring should be available during the fall of the 1975-76 school year. Presently, the simplest, and perhaps the most meaningful, method of scoring is to allow the students to score their own. Preparing students for this entails writing the names of the nine (9) construct areas the Instrument covers on the chalkboard or overhead projector and reading these names. The procedures listed here have been tried successfully in other classrooms and are easy to follow. Each student should be given pages headed "Learning Styles Worksheet," and "Learning Style Chart." The procedure for completing the worksheet follows:

1. The numbers listed under each of the nine (9) construct areas designate the statements on the Instrument which measure that particular style. For example, statements 5, 13, 21, 29, 37 all measure the visual language style of learning.
2. To determine the score for each style, do the following:
  - a. Look up the response given for each statement and write it in the right-hand column.
  - b. Total the numbers.
  - c. Multiply the sum of the numbers by two (2).

This is the score:

Example:

Visual Language

5 - 3

754

|        |                           |
|--------|---------------------------|
| 13 - 4 |                           |
| 21 - 4 | Suppose these are the     |
| 29 - 3 | responses given on the    |
| 37 - 4 | Learning Style Instrument |

TOTAL  $18 \times 2 = 36$

3. Figure the score for each of the nine (9) areas. If the score is 33 through 40, the style is considered to be a major learning style for the student. From 20 through 32 is a minor learning style for the student, and below 20 indicates the student uses this style to a negligible extent.

After computing each score, the students graph the results on the page headed "Learning Style Chart." The completed sample of the "Learning Style Chart" is on the following page.

Style usage may be defined as:

1. Major: The student prefers this mode of learning, feels comfortable with it, and uses it for important (to the student) learning. A student does not necessarily have one and only one preferred style.
2. Minor: The student uses this mode but usually as a second choice or in conjunction with other learning styles.
3. Negligible: The student prefers not to use this if other choices are available.

The following are descriptions of learning styles which can be found in every learner to a major, minor, or negligible extent. These descriptions will help you evaluate a student's learning style on the basis of observation. The CITE Learning Styles Instrument is a tool which facilitates the evaluation of these learning styles. Each of the nine (9) style areas of the CITE Instrument are described here as if the student were a major in that particular style.

1. Auditory Language: This is the student who learns from hearing words spoken. You may hear him vocalizing or see his lips or throat moving as he reads, particularly when he is striving to understand new material. He will be more capable of understanding and remembering words or facts that he could only have learned by hearing.

2. Visual Language: This is the student who learns well from seeing words in books, on the chalkboard, charts, or workbooks. He may even write words down that are given to him orally, in order to learn by seeing them on paper. He remembers and uses information better if he has read it.
3. Auditory Numerical: This student learns from hearing numbers and oral explanations. He may remember phone and locker numbers with ease, and be successful with oral numbers, games, and puzzles. He may do just about as well without his math book, for written materials are not as important. He can probably work problems in his head. You may hear him saying numbers to himself or see his lips moving as he reads a problem.
4. Visual Numerical: This student has to see numbers on the board, in a book, or on a paper, in order to work with them. He is more likely to remember and understand math facts if he has seen them. He doesn't seem to need as much oral explanation.
5. Auditory-Visual-Kinesthetic Combination: The A-V-K student learns best by experience--doing, self-involvement. He definitely needs a combination of stimuli. The manipulation of material along with the accompanying sight and sounds (words and numbers seen and spoken) will make a big difference to him. He may not seem to be able to understand or be able to keep his mind on work unless he is totally involved. He seeks to handle, touch, and work with what he is learning. Sometimes just writing or a symbolic wiggling of the fingers is a symptom of the A-V-K learner.
6. Individual Learner: This student gets more work done alone. He thinks best and remembers more when he has learned by himself. He cares more for his own opinions than for the ideas of others. You will not have much trouble keeping this student from over-socializing during class.
7. Group Learner: This student strives to study with at least one other student, and he will not get as much done alone. He values others' opinions and preferences.  
  
Group interaction increases his learning and later recognition of facts. Class observation will quickly reveal how important socializing is to him.
8. Oral Expressive: This student can easily tell you what he knows. He talks fluently, comfortably and seems to be able to say what he means. You may find he knows more than his tests show, after talking to him about his work. He is probably not shy about giving reports or talking to the teacher or classmates. The

muscular coordination involved in writing may be difficult for him. Organizing and putting thoughts on paper may be a too slow and tedious task for this student.

9. Written Expressive: This student can write fluent essays and good answers on tests to show what he knows. He feels less comfortable, perhaps even stupid, when he has to give oral answers. His thoughts are better organized on paper than when they are given orally.

The CITE Learning Styles Instrument is an attempt to help teachers become more familiar with their student's preference for learning and to make teachers more aware of the need to accommodate the variety of preferences exhibited by any class of students. Should this instrument show no significant data on a student, or should the teacher feel the data is not accurate, then further investigation should be made. Private talks between the teacher and student should follow. The student should be given choices of a variety of ways of learning and these choices noted on the student's records and compared with his/her Learning Style Chart. Personal observation by the teacher is also a helpful way of determining the student's learning style. If adequate diagnostic or counseling service is available, the use of these services will help in determining learning style. Each of these other methods mentioned requires a great deal of time, effort, expertise, and one-to-one work. Sometimes the classroom teacher runs short on one or more of these. That is where the CITE Learning Styles Instrument is meant to help.

## LEARNING STYLE

**INSTRUCTIONS:** Place all answers on the answer sheet. Circle the number which agrees with how you feel about the statement. A 4 indicates you agree that the statement is most like you, and a 1 indicates that the statement is least like you.

1. When I make things for my studies, I remember what I have learned better.
2. Written assignments are easy for me to do.
3. I learn better if someone reads a book to me than if I read silently to myself.
4. I learn best when I study alone.
5. Having assignment directions written on the board makes them easier to understand.
6. It's harder for me to do a written assignment than an oral one.
7. When I do math problems in my head, I say the numbers to myself.
8. If I need help in the subject, I will ask a classmate for help.
9. I understand a math problem that is written down better than one I hear.
10. I don't mind doing written assignments.
11. I remember things I hear better than I read.
12. I remember more of what I learn if I learn it when I am alone.
13. I would rather read a story than listen to it read.
14. If feel like I talk smarter than I write.
15. If someone tells me three numbers to add I can usually get the right answer without writing them down.



**Business Computer Programming I**  
**APPENDIX G: Special Needs**

16. I like to work in a group, because I learn from the others in my group.
17. Written math problems are easier for me to do than oral ones.
18. Writing a spelling word several times helps me remember it better.
19. I find it easier to remember what I have heard than what I have read.
20. It is more fun to learn with classmates at first, but it is hard to study with them.
21. I like written directions better than spoken ones.
22. If homework were oral, I would do it all.
23. When I hear a phone number, I can remember it without writing it down.
24. I get more work done when I work with someone.
25. Seeing a number makes more sense to me than hearing a number.
26. I like to do things like simple repairs or crafts with my hands.
27. The things I write on paper sound better than when I say them.
28. I study best when no one is around to talk or listen to.
29. I would rather read things in a book than have the teacher tell me about them.
30. Speaking is a better way than writing if you want someone to understand what you really mean.
31. When I have a written math problem to do, I say it to myself to understand it better.
32. I can learn more about a subject if I am with a small group of students.
33. Seeing the price of something written down is easier for me to understand than having someone tell me the price.

- 34. I like to make things with my hands.
- 35. I like tests that call for sentence completion or written answers.
- 36. I understand more from a class discussion than from reading about a subject.
- 37. Spelling and grammar rules make it hard for me to say what I want to in writing.
- 38. It makes it easier when I say the numbers of a problem to myself as I work it out.
- 39. I like to study with other people.
- 40. When the teachers says a number, I really don't understand it until I see it written down.
- 41. I understand what I have learned better when I am involved in making something for the subject.
- 42. Sometimes I say dumb things, but writing gives me time to correct myself.
- 43. I do well on tests if they are about things I hear in class.
- 44. I can't think as well when I work with someone else as when I work alone.



## LEARNING STYLES ANSWER SHEET

NAME:

DATE:

GRADE

Circle the number which agrees with how you feel about the statement. A 4 indicates you agree that the statement is most like you, and a 1 indicates that the statement is least like you.

|     | LEAST<br>LIKE ME |   |   | MOST<br>LIKE ME |
|-----|------------------|---|---|-----------------|
| 1.  | 1                | 2 | 3 | 4               |
| 2.  | 1                | 2 | 3 | 4               |
| 3.  | 1                | 2 | 3 | 4               |
| 4.  | 1                | 2 | 3 | 4               |
| 5.  | 1                | 2 | 3 | 4               |
| 6.  | 1                | 2 | 3 | 4               |
| 7.  | 1                | 2 | 3 | 4               |
| 8.  | 1                | 2 | 3 | 4               |
| 9.  | 1                | 2 | 3 | 4               |
| 10. | 1                | 2 | 3 | 4               |
| 11. | 1                | 2 | 3 | 4               |
| 12. | 1                | 2 | 3 | 4               |
| 13. | 1                | 2 | 3 | 4               |
| 14. | 1                | 2 | 3 | 4               |
| 15. | 1                | 2 | 3 | 4               |
| 16. | 1                | 2 | 3 | 4               |
| 17. | 1                | 2 | 3 | 4               |
| 18. | 1                | 2 | 3 | 4               |
| 19. | 1                | 2 | 3 | 4               |

Business Computer Programming I  
APPENDIX G: Special Needs

|     |   |   |   |   |
|-----|---|---|---|---|
| 20. | 1 | 2 | 3 | 4 |
| 21. | 1 | 2 | 3 | 4 |
| 22. | 1 | 2 | 3 | 4 |
| 23. | 1 | 2 | 3 | 4 |
| 24. | 1 | 2 | 3 | 4 |
| 25. | 1 | 2 | 3 | 4 |
| 26. | 1 | 2 | 3 | 4 |
| 27. | 1 | 2 | 3 | 4 |
| 28. | 1 | 2 | 3 | 4 |
| 29. | 1 | 2 | 3 | 4 |
| 30. | 1 | 2 | 3 | 4 |
| 31. | 1 | 2 | 3 | 4 |
| 32. | 1 | 2 | 3 | 4 |
| 33. | 1 | 2 | 3 | 4 |
| 34. | 1 | 2 | 3 | 4 |
| 35. | 1 | 2 | 3 | 4 |
| 36. | 1 | 2 | 3 | 4 |
| 37. | 1 | 2 | 3 | 4 |
| 38. | 1 | 2 | 3 | 4 |
| 39. | 1 | 2 | 3 | 4 |
| 40. | 1 | 2 | 3 | 4 |
| 41. | 1 | 2 | 3 | 4 |
| 42. | 1 | 2 | 3 | 4 |
| 43. | 1 | 2 | 3 | 4 |
| 44. | 1 | 2 | 3 | 4 |

## LEARNING STYLES WORKSHEET

### VISUAL LANGUAGE

5 \_\_\_\_\_  
13 \_\_\_\_\_  
21 \_\_\_\_\_  
29 \_\_\_\_\_  
37 \_\_\_\_\_

Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

### AUDITORY NUMERICAL

7 \_\_\_\_\_  
15 \_\_\_\_\_  
23 \_\_\_\_\_  
31 \_\_\_\_\_  
39 \_\_\_\_\_

Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

### VISUAL NUMERICAL

9 \_\_\_\_\_  
17 \_\_\_\_\_  
25 \_\_\_\_\_  
33 \_\_\_\_\_  
41 \_\_\_\_\_

Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

### KINESTHETIC-TACTILE

1 \_\_\_\_\_  
18 \_\_\_\_\_  
26 \_\_\_\_\_  
34 \_\_\_\_\_  
42 \_\_\_\_\_

Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

### AUDITORY LANGUAGE

3 \_\_\_\_\_  
11 \_\_\_\_\_  
19 \_\_\_\_\_  
36 \_\_\_\_\_  
44 \_\_\_\_\_

Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

### SOCIAL INDIVIDUAL

4 \_\_\_\_\_  
12 \_\_\_\_\_  
20 \_\_\_\_\_  
28 \_\_\_\_\_  
45 \_\_\_\_\_

Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

Business Computer Programming I  
APPENDIX G: Special Needs

**SOCIAL GROUP**

8 \_\_\_\_\_  
16 \_\_\_\_\_  
24 \_\_\_\_\_  
32 \_\_\_\_\_  
40 \_\_\_\_\_  
Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

**EXPRESSIVE-ORAL**

6 \_\_\_\_\_  
14 \_\_\_\_\_  
22 \_\_\_\_\_  
30 \_\_\_\_\_  
38 \_\_\_\_\_  
Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

**EXPRESSIVE-WRITTEN**

2 \_\_\_\_\_  
10 \_\_\_\_\_  
27 \_\_\_\_\_  
35 \_\_\_\_\_  
43 \_\_\_\_\_  
Total \_\_\_\_\_ x 2 = \_\_\_\_\_ (Score)

33-40 Major Learning Style \_\_\_\_\_

20-32 Minor Learning Style \_\_\_\_\_

20 Negligible \_\_\_\_\_

76.4

## Teaching Strategies

Once the student's vocational goals, learning capability, and preferred learning style have been identified, the educator can begin to experiment with a variety of teaching techniques and instructional materials. When determining the teaching technique and instructional material best suited to meet a student's unique needs, it is important for the educator to be patient, open-minded, and flexible. Not every instructional technique and/or material can be used successfully with each learner (Sarkees & Scott, 1985). In addition, the educator should not hesitate to enlist the support of other instructors, school counselors, student aides, paraprofessionals, volunteers from the community, itinerant teachers, and/or interpreters. This pooling of efforts can result in increased student learning because of the variety of approaches and personal attention which can be brought to the classroom or laboratory (Harrington & Fardig, 1982).

The following are general guidelines and instructional techniques which an educator can use to help meet a special needs learner's unique needs and learning style:

1. Orientation. Students with special needs may need additional orientation with regard to: (a) the relevancy of their individualized goals, (b) class procedures, (c) safety, etc.
2. Flexible pacing. Students with special needs can exhibit different rates of learning. The educator should provide flexibility in pacing of instruction and assignments to meet the various needs of learners. In general, students with special needs should be individually paced to maintain their progress and interest, yet avoid the frustrations and failure undue pressure may cause (Harrington & Fardig, 1982; Sarkees & Scott, 1985).
3. Demonstration. Utilization of the demonstration method can be very effective in introducing a new skill to a learner with special needs.
4. Breaking material into small distinct steps. As a general rule, it is important for the educator to break a learning concept into small distinct steps. In this way, the educator can best identify where the student is experiencing difficulty.
5. Positive environment and feedback. In general, students with special needs lack confidence and a positive self-concept. This is primarily due to their not having had much success in life. In addition, the few successes special learners do have usually result in little, if any, positive reinforcement (i.e., praise, reward, and/or encouragement). Therefore, it is incumbent upon educators to create a positive,

nonthreatening learning environment for their students and to make generous use of positive feedback when the student experiences success. Constructive criticism should not be avoided; however, it should be given in a positive manner (i.e., students should understand that their behavior or product is what's being criticized and not their worth as a human being). Finally, all students must learn that mistakes and constructive criticism are a normal and expected part of life (Harrington & Fardig, 1982).

6. Use of peer tutors, college students, and volunteers. Use of peer tutors, college students, and volunteers can greatly assist a vocational educator's work with special learners by: (a) helping with individualized instruction, (b) providing positive role models, (c) increasing student participation, (d) increasing the special learner's success experiences, (e) increasing the student's confidence and self-esteem, and (f) improving the special learner's social and academic skills (Asselin & Anderson, 1986). In addition, peer tutors, college students, and volunteers can receive teaching experience and the satisfaction of helping individuals with special needs progress.
7. Flexible grouping. The educator can integrate special needs learners into groups of various sizes (according to the student's goals, interests, or ability level) and utilize peer tutors, college students, and volunteers for assistance. According to Sarkees & Scott, grouping students has the following advantages: (a) it allows the teacher more time to work among the students, (b) it allows the teacher to individualize instruction, tasks, and assignments to meet the needs of students in a specific group, (c) it provides special needs learners with an opportunity to work with peers and establish positive interpersonal and working relationships with them, (d) it serves as a motivating factor for special needs learners because the individualized nature of the group work does not become threatening or frustrating but allows students an opportunity to work at their own pace through their strongest learning style, (e) it allows teachers to arrange for instruction for the entire class, small group discussions, project groups, and small remedial sessions so that instruction will be diversified and more meaningful for all students; and (f) it allows the teacher to modify the task or assignment for a small group of students with specific needs, as in the case of learners who have a physical disability (Sarkees & Scott, 1985).
8. Providing field trips and job site visits. Scheduled field trips and job site visits can give all learners a realistic first-hand look into the real world of work. By "shadowing" workers in a particular occupational area, students can gain helpful insights into their own career decision. In addition, a field trip or job site visit can

be very motivational in that the student can readily observe how his/her vocational training will be used in the world of work (Sarkees & Scott, 1985).

9. Use of media. Utilizing media can embellish instruction by: (a) assisting the teacher in presenting the proficiency or skill requirements for a specific job, (b) reinforcing concepts presented in a classroom, laboratory, or demonstration activity, (c) providing a vehicle for special needs learners to show what they have learned, (d) providing a valuable source of remedial assistance, (e) providing simulated learning experiences, (f) allowing special needs students to learn through their strongest style, and (g) adding variety to the instructional program. Examples of media include: computer assisted instruction (CAI), 16 mm films, film strips, flip charts, overhead transparencies, tape recorders, records, posters, magazines, and diagrams (Sarkees & Scott, 1985).
10. Use of student-teacher contracts. Student-teacher contracts can be excellent tools in helping learners with special needs achieve their vocational goals. The following guidelines should be used in developing and implementing a student-teacher contract: (a) decide on the task or project, (b) specifically state the requirements, (c) determine the minimum acceptable accuracy, (d) set a time limit--initially, short assignments and short time periods are usually best, (e) identify consequences, (f) discuss the contract with student and support personnel, (g) have all parties sign the contract, (h) positive reinforcement should be given if the terms of the contract are satisfactorily met, (i) uncompleted contracts should be analyzed by the teacher and student to determine the causes, and (j) develop new contracts cooperatively (Sarkees & Scott, 1985).

The following is a list of specific teaching strategies which educators can utilize when addressing the unique needs of handicapped students (Sarkees & Scott, 1985; Stern & Gathercoal, 1986; Tindall; 1980):

**Important Note:** It is very important to note that many of these teaching strategies could be used for all students. Appearing later in this Overview is a listing of instruction strategies which could be generic to all special needs students.

### **Instructing Students Who Are Mentally Impaired**

Involve as many different senses as possible in the learning process.

Use concrete specific terms rather than abstract terms. If necessary, have the student repeat the necessary information to you.



When giving written directions use lists instead of paragraphs. With regard to the student who can barely read, contact the remedial reading staff and arrange reading instruction. This instruction should focus on required reading for your course and that which is vital for employment.

Teach by demonstration (whenever possible). Don't hesitate to use hand-over-hand assistance in demonstrating how to use a tool or machine.

Allow time to practice machine or tool processes.

Break down complex operations into smaller tasks.

With regard to feedback, let the learner know what is wanted, and whether or not the student is achieving it. Give lots of positive reinforcement and encouragement. Nonverbal feedback is also helpful (i.e., touching the student or task, pointing, blocking or stopping an error as it is developing, and guiding the learner's movements). In addition, provide specific feedback with regard to the student's work. For example, the educator should say, "This piece was cut exactly to the right size," or "You put these two parts together correctly," or use Mark Gold's expression "Try another way" (Gold, 1975).

Build the student's confidence by: (a) utilizing activities that will assure success, (b) assigning special tasks which will communicate with the educator's confidence in the student, (c) giving students an opportunity to correct their work and hand in a good project or assignment, and (d) displaying the student's good work.

In helping students with their ability to make correct discriminations, utilize Gold's "match to sample" technique in which the student is provided an example and a number of choices. The student will choose the correct alternative based upon the characteristics of the sample (Gold & Scott, 1971).

With regard to helping the student make correct discriminations when there are minimal differences between the stimuli, it may be necessary to: (a) present the choices in an easy form at first, (b) exaggerate attributes, (c) make the choices gradually harder, or (d) add and fade out color cue, add and then gradually remove large cue differences, or combine these two approaches.

Retention/memory can be improved by: (a) being systematic and presenting one new piece of learning at a time, (b) relating new material to familiar material, and keeping it practical and concrete, and (c) utilizing training techniques such as the total task method, chaining, modeling/demonstrating, and the structured overview (see Tindalls' *Puzzled About Educating Special Needs Students*).



"Generalizing," or having the student apply what has been learned to the contexts, can be improved by: (a) using examples from different relevant contexts and (b) practicing in settings similar to or exactly like those that will be encountered on the job.

### **Instructing Students Who Are Hearing-Impaired**

In private, discuss with the student his/her ability to communicate with you (lip/speech reading, use of hearing aid, writing, signing, and total communication). An open discussion with the student which demonstrates your concern as an educator is an excellent beginning in building rapport.

Body language communicates messages too. Remember not to give mixed messages with your body language. Also, be alert to the body message the student is giving you (i.e., is he/she communicating tension, nervousness, avoidance, or confusion?).

If you will be working with an interpreter, do not instruct the deaf student as if he/she were a third party. Speak directly to the student. Also, remember that the interpreter is ethically bound to interpret everything you say. Therefore, sensitive or other issues concerning the student is best addressed in a meeting before or after class. If you are using media where the room must be darkened, focus a small lamp on the interpreter.

Maintain good eye contact with the student while speaking.

Speak slowly and clearly, pronouncing each word, but without exaggerating or over pronouncing.

Do not stand in front of a window while speaking. The light should be shining on your face and not the student's.

Try to rephrase a thought rather than saying the same words over again.

Encourage the student to read ahead on class material.

Try to keep the noise level down as much as possible in the class/laboratory.

Utilize handouts, the chalkboard, and other media.

Install a light to indicate when dangerous machinery is on.

### **Instructing Students Who Are Speech-Impaired**

Always maintain eye contact and nod your head while the student is talking. This will help the student build confidence.

Work cooperatively with speech therapist and/or communication specialist to determine which technique will help most in assisting these students to communicate more effectively.

Encourage peer tutoring and small group activities to help the speech-impaired student develop interpersonal relationships with the other students.

Always allow sufficient time for the student to complete his/her sentence.

Try not to act uncomfortable when these individuals are communicating with you (i.e., discontinuing eye contact, shifting your position, clearing your throat). Your discomfort can be very discouraging to the student.

Do not require these individuals to present oral reports until they feel comfortable in the classroom environment.

Model correct sentence structure for the speech-impaired student so he/she can follow your example.

Develop a list of technical terms used in your class and share these terms with the speech specialist to help the student become more confident pronouncing terms used in class.

### **Instructing Students Who Are Visually Impaired**

Orient the student to the classroom or laboratory before the course begins. Consistency and orderliness in the physical environment is very important to the visually impaired student.

Arrange seating so the blind student will be able to hear the instruction and to allow for the best distance and lighting conditions for the partially sighted student.

Provide as many opportunities as possible for the visually impaired student to observe closely or to touch the materials.

Give materials and handouts to the student ahead of time. Also, a student peer can assist the visually impaired student by making a carbon copy or a photocopy of his/her notes. Also, when possible, the educator can make a copy of his/her teaching notes. In this way, the visually impaired student can review the materials with a tutor or have them transcribed into braille or taped.

It is important to gain and maintain the student's attention. Approval or disapproval will need to be verbalized.

Remember there is no need to talk louder to a visually impaired student.

Rehabilitation counselors or special needs teachers can be useful resources in helping the student use commercial or modified learning aids or tools.

Tests may need to be prepared in large print, braille, on cassette tapes, or converted into exercises using the sense of touch. Also, it may be necessary to allow the student extra time to complete the test.

The tape recorder is the most frequently used technique for notetaking and preparing modified learning materials for the visually impaired student.

The following is a sample of specially designed machines now available for the visually impaired student: (a) OPTACON (Optical to Tacile Converter) allows students to read printed material, (b) STEREO-TONER is similar to the OPTACON but converts the printed word into a tone pattern, (c) OPTOPHONE and VISOTACTOR convert print into sound, and (d) KURZWEL READING MACHINE which converts standard English text on a computer into easily comprehended speech.

A number of general tools or aids which can assist the student in reading or writing lessons for vocational courses follow: (a) braille, (b) large print, (c) tape recorders, (d) talking books, (e) raised letters, (f) slate and stylus (used to make notes), (g) braille writer (a 6-key manually operated machine), (h) typewriters, (i) braille verifier (types braille and regular copy simultaneously), and (j) signature guides.

Aids for mathematics and computer programming courses follow: (a) abacus design for the blind, (b) raised clock faces, (c) wire geometrical forms, (d) braille rulers, (e) raised-line checkbooks, (f) raised-line boards, (g) braille dominoes, (h) braille flash cards, (i) cubes with raised braille dots used for simple mathematical problems, (j) raised-line paper or graphs, (k) computer terminals that use braille printers, or those that type braille and conventional type simultaneously, and (l) talking calculators.

Aids used frequently in other vocational courses include: (a) braille maps, charts, diagrams, (b) self-threading needles, (c) wooden or cardboard models of specimens that other students see through microscopes, (d) braille micrometers and other shop tools that have special raised markings, (e) braille demo tapes to mark lab equipment and models, (f) light probes with audible sounds that check the operating status of machines, and (g) guides or templates to help students align objects properly.

### **Instructing Students Who Are Seriously Emotionally Disturbed**

Consult with the student's parents, special education teacher, or counselor to determine whether any particular teaching techniques will help or hinder learning.

Try to understand the student. Disturbing behaviors are the outward sign of inner turmoil.

Go easy on bad days and avoid direct confrontations that would aggravate the situation.

Communicate acceptance to the student and a willingness to help him or her succeed in class.

Provide immediate feedback and reinforcement.

Establishing a rapport and trust with the student is important.

In the beginning of the semester, help the student gain confidence by giving him or her lessons in which success can be achieved in a series of small steps.

A measure of privacy when the student attempts a new task can ease his or her embarrassment at any awkwardness or ineptness.

Allow the student to correct mistakes prior to turning in an assignment or project to ease any pressure.

Help the student not to overreact to failure and praise him or her for continuing to try.

Cue appropriate behavior in a quiet, unemotional manner.

In resolving discipline problems: (a) do so in privacy, (b) focus on objective facts in a calm, supportive tone, (c) actively listen to the student, (d) after listening to the student, reflect on what has been said by telling the student what you've heard, (e) ask the student if he or she has any ideas about solving the problem (encourage the student to be specific), and (f) conclude with a clear understanding of the changes, time frame, and benefits which are expected.

In handling aggressive behavior: (a) try to remain calm and model the controlled behavior which you desire, (b) intervene by either separating the individuals involved and having them work on another activity or having them take a time-out in separate corners of the room, (c) if necessary, send an uninvolved student to the office to get help, either in talking to the students or in managing the class while the teacher talks to them, and (d) return to normal classroom procedure as quickly as possible.

Access professional colleagues in special education evaluation, counselling, or therapy when the student's behavior appears to be in need of additional attention.

### **Instructing Students Who Are Orthopedically Impaired**

In general, students with physical handicaps will be intellectually capable and able to participate in vocational programs.

Make the necessary modifications in classrooms, laboratories, equipment, building accessibility, and lavatories.

For notetaking the student can: (a) use a tape recorder, (b) have a classmate slip a carbon copy into his/her notebook to duplicate notes, and/or (c) have the instructor prepare an outline of the lecture.

For exams the instructor can: (a) give oral exams, (b) have the student dictate answers into a tape recorder or to another student, or (c) utilize a typewriter/computer if the student is able to type/keyboard.

Give the student more time to complete homework assignments. Also, giving the student advance notification of assignments is helpful.

In the laboratory, pair the student with one or more capable students in the class.

Enlarged copies of frequently used charts and diagrams posted near the student's desk eliminate the necessity of paging through a book.

### **Instructing Students Who Are Other Health-Impaired**

Allow the students sufficient time for rest.

Seat the student in the most convenient place for easy access to necessary facilities (hall, lavatory, chalkboard, laboratory work stations).

Allow the students to work on program requirements at his or her own pace.

Tape lectures so the student can listen to them at home.

Work cooperatively with the health team in order to learn more about the student's special needs.

Utilize community resources personnel (social workers and itinerant teachers for home-bound students) for delivery of materials, assignments, and handouts.

### **Instructing Students Who Have Specific Learning Disabilities**

Use visuals and demonstrations as often as possible.

Use concrete examples whenever possible.

Provide student with an outline of what will be covered in class.

Make sure all printed materials are clear and readable.

If the student has a visual perception problem, allow him or her to tape-record assignments.

Allow students to take tests orally or in a mode that complements their performance.

Present new information in small amounts.

Make sure that writing assignments are relevant to the content of the program.

Allow students a longer period of time to complete assignments, if necessary.

List assignments in steps.

Task analysis can be used to lower the frustration level of these learners.

Programmed instruction materials can be very effective with these learners.

The following is a list of specific teaching strategies which can be very useful when working with students who have limited English proficiency (Sarkees & Scott, 1985):

Recognize the student's cultural background and allow for the student's individual differences.

Encourage these students to bring dictionaries to class.

Use demonstrations and hands-on experience as much as possible.

Speak slowly and use simple vocabulary and grammar.

Use bright felt tip markers to mark important words, passages, or illustrations.

If possible, use a translator for important concepts.

Work closely with LEP support personnel.

Keep homework assignments short.

Check for comprehension by asking the student what you have said.

Use simple sentences when writing test items so the student can concentrate on answering the question rather than deciphering it.



And finally, with regard to the disadvantaged population, typically these individuals dislike school, lack motivation, and have a low self-concept or esteem. Therefore, an educator's personal commitment and high expectations for the student may be two of the biggest single factors which will make a positive difference in this individual's life. Additionally, an educator's good example (i.e., being patient, understanding, well organized, and a good manager) can be extremely valuable in motivating and positively shaping the student's behavior (Baca, 1981). Vocational education can offer a great deal to the disadvantaged student in terms of teaching academic and social skills relative to the student's potential employment (Baca, 1981). The following are additional techniques or teaching strategies which can be very useful when working with disadvantaged students (Baca, 1981):

The educator should seek to understand the student's educational and social background through: (a) assessment information, (b) interaction with former teachers, (c) discussions with counselors, and (d) visits with parents to ascertain the student's social limitations and home situation.

Basic math, communication, and social skills should be reinforced. Math skills can be reinforced through hands-on learning activities. Communication skills can be improved by tactfully correcting poor grammar, when appropriate, and by providing resources that encourage reading in job-related interests. Social skills such as getting along with people, grooming, proper dress, and respect for property and authority should be reinforced on a continual basis.

The importance of finishing school should be continually stressed. Relate acquisition of job skills to earn income.

Reinforce the student's feelings of self-worth by having high expectations for the student and by providing immediate reinforcement for work done well.

Simulate realistic job situations. All classroom and laboratory experiences should relate to employment in the occupation being studied.

Disadvantaged students (as all students) will typically react to the learning environment in a responsible and mature manner when they are treated with respect and as adults by the educator.

Classroom organization and discipline (being firm but not overbearing) are especially important. Generally, disadvantaged students want and need structure and discipline within their lives. Usually, these individuals will be responsive to and respect teachers who are firm, but sensitive.

Help the student develop employability and job-coping skills (i.e., skills associated with finding, applying for, and retaining a job).

Help the student establish good work habits by keeping him or her involved with meaningful learning experiences. Each activity should have concrete learning objectives.

Disadvantaged students may need more encouragement to stay on task; however, they still should be required to complete all assignments (just as they will be required to do on the job).

Present instructions and information in clear, logical steps--building from the simple to the complex.

Technical terms should be communicated clearly and concisely and written in language which the students will understand.

Incorporate a variety of instructional methods in the curriculum (i.e., demonstrations, resource personnel, field trips, peer tutors, role playing, audio-visual aids, and on-site work experiences).

Present supplemental materials when appropriate. If such material is relevant to the acquisition of job skills, the student will not be humiliated by simplified material.

Highlight important parts of the student's text with a light marker to prevent the student from feeling overwhelmed with a larger volume of reading material.

Progress charts can be utilized to motivate disadvantaged students to master new skills.

Utilize flash cards (teacher-made or commercial) to reinforce new terms and the use of tools and equipment.

When teaching special needs students, it is very important that instructors do not limit their instructional techniques. Therefore, the following abstract of instructional techniques or strategies has been prepared which could be considered generic to all special needs populations.

- Involve as many different senses as possible
- Use concrete terms
- Have students repeat important information
- Use lists rather than paragraphs for written directions
- Teach by demonstration
- Use hand-over-hand assistance when necessary



- Allow practice time
- Break complex operations into smaller tasks
- Provide immediate feedback
- Build confidence
  
- Help student make correct discriminations
- Help students with their retention/memory
- Help student generalize knowledge
- Build rapport
- Use appropriate body language
- Use appropriate techniques when working with an interpreter
  
- Maintain good eye contact
- Speak slowly and clearly
- Rephrase thoughts when necessary
- Encourage the student to read ahead
- Keep the noise level down
- Utilize various forms of media
- Use lights to indicate when dangerous machinery is on
- Work cooperatively with therapists
- Encourage peer tutoring
- Try not to act uncomfortable around special needs students
- Help students feel comfortable
- Model correct sentence structure
- Develop a list of technical terms for students and/or therapists
  
- Orient students before the course begins
- Provide handouts ahead of time
- Maintain student's attention
- Use tape recorders for note taking
- Try to understand the student
- Communicate acceptance

- Develop success experiences for the student
- Use appropriate behavioral management techniques
- Facilitate the students with their notetaking
- Facilitate the students with exams
- Pair students
- Enlarge charts and diagrams
- Use clear readable visuals frequently
- Provide an outline of material which is being taught
- Writing assignments should be relevant
- Use task analysis when necessary
- Allow for individual differences
- Encourage students to bring a dictionary
- Use felt-tip marker to highlight important parts
- Do not make homework assignments too lengthy
- Use simple sentences when writing test items
- Reinforce basic education and work skills
- Reinforce the importance of finishing school
- Reinforce the student's feelings of self-worth
- Simulate realistic job situations
- Treat all students with respect and dignity
- Provide structure and organization
- Utilize progress charts
- Utilize flash cards

### **Remediation in the Basic Education Skills**

It is very important to note that some students may need to acquire one or more basic education skills (i.e., mathematics, reading, writing, speaking, listening, interpersonal relations, and/or reasoning skills) to succeed in a vocational program. In addition, many employers will not hire an individual if he or she lacks these skills. Therefore, instruction

in basic education skills may be necessary by the vocational educator. For information concerning basic or generalized skills instruction, see Module 8.

### **Brown's 3-C Model**

Brown's 3-C Model helps tie together many of the educational aspects discussed in this Overview. In general, Brown suggests that when delivering instruction to learners with special needs, the instructor can enhance the student's success rate and reduce feelings of frustration by considering the following 3-Cs of instruction: (a) correction, (b) compensation, and (c) circumvention (Brown & Dayser, 1982):

*Correction.* Providing feedback when the student's skills are being developed, thereby correcting inappropriate behaviors immediately after they appear. Employing remedial reading and math services to enhance the student's ability to function within the program.

*Compensation.* Helping the student realize that each task represents only one of many activities contained in the course and that weaknesses in one area can often be offset by strengths in other areas. It should be communicated to the student that some tasks are inherently difficult and that he/she should not expect to achieve success immediately. Further efforts to compensate for the student would be to modify or adjust the delivery of instruction to extend over segments of several class periods. In addition, reinforcement during the early stages of instruction should be amply provided.

Note: Basically, "compensation" involves the instructor's modifications to the curriculum, learning environment, and instruction which has been discussed in this Overview.

*Circumvention.* If the vocational program is way out of reach for a student, for example, due to a course requirement of a relatively high level of reading skill, it may become necessary for the student to "circumvent" this difficulty by discussing alternate training programs with a vocational counselor.

### **Summary**

Modification of curriculum, learning environment, and instruction can be an important key to successful employment training for special needs students. It requires that the instructor know his/her content completely, but the benefits to all students will likely be extraordinary.

## BIBLIOGRAPHY

Asselin, S.B. & Anderson, A.G. (1986). Peer tutors: An idea that works. *The Journal For Vocational Special Needs Education*. Vol. 8, pp. 21-24.

Baca, P. (1981). For vocational teachers: Some helpful strategies for working with disadvantaged students. *The Journal for Vocational Special Needs Education*. Vol. 4, pp. 3-4, 22.

Bell, J.A. (1984). Modify the learning environment for special students. Module L-5. Professional Teacher Education Module Series. The National Center for Research in Vocational Education. The Ohio State University, Columbus, Ohio.

Brolin, D.E. (1976) *Vocational preparation of retarded citizen*. Columbus: Charles E. Merrill Publishing Company.

Brown, J.M. & Kaysar, T.F. (1982). The transition of special needs learners into postsecondary vocational education. Minnesota Research and Development Center for Vocational Education. University of Minnesota.

Choate, J.S.; Bennett, T.Z.; Enright, B.E.; Miller, L.J.; Poteat, J.A.; & Rankes, T.A. (1987). *Assessing and Programming Basic Curriculum Skills*. Massachusetts: Allyn and Bacon, Inc.

Cobb, R.B. & Kingbury, D.E. (1985). The special needs provisions of the Perkins Act. *Vocational Education*, 60(4).

Gleiss-Monar, K. & Sitlington, P.L. (1983). Modifying employment-related facilities and instructional techniques for the special need learners. Departments of Special Education and Organization and Occupational Development. Indiana University, Bloomington.

Goh, S.R. & Sitlington, P.L. (1983). Curriculum modifications for special needs learners in employment related-training. Departments of Special Education and Organization and Occupational Development. Indiana University, Bloomington.

Gold, M. (1975). Symposium on applying attribution theory to social problems: Vocational skill functioning of the severely retarded. Institute for Child Behavior and Development. University of Illinois at Urbana-Champaign.

Gold, M. & Scott, K.G. (1971). Discrimination learning. In W.B. Stephen's (Ed.). *Training the Developmentally Young*. New York: John Day.

Harrington, L.G. & Fardig, G.E. (1982). Use instructional techniques to meet the needs of exceptional students. Module L-7. Professional Teacher Education Module Series. The National Center for Research in Vocational Education. The Ohio State University, Columbus, Ohio.

Hill, P. (Unpublished). Vocational special needs handbook. Texas Education Agency.

Jones, D.M. (1977). *Curriculum Targets in the Elementary School*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.

King-Fitch, C.C. (1984). Assist students in improving their math skills. Module M-5. Professional Teacher Education Module Series. The National Center for Research in Vocational Education. The Ohio State University, Columbus, Ohio.

Kok, M.; Jernigan, J.; & Hull, M. (1978). Curriculum development instructional module. Project Encounter. Texas A&M University, College Station, Texas.

Lippmann, G.K. & Porter, G.C. (1976). How to establish competency model programs for the delivery of vocational services to the severely multihandicapped. Copyright by Lippmann/Porter. Austin, Texas.

Parnell, D. (1973). Career education and the school curriculum. In L. McClure's, and C. Buan's (Eds.), *Essays on Career Education*. Portland, Oregon: Northwest Regional Education Laboratory.

Phelps, L.A. & Lutz, R.J. (1977). *Career Exploration and Preparation for the Special Needs Learner*. Boston: Allyn and Bacon.

Pine, G.J. & Boy, A.V. (1977). *Learner Centered Teaching. A Humanistic View*. Denver, Colorado: Love Publishing Company.

Sarkees, M.D. & Scott, J.L. (1982). *Vocational Special Needs* (2nd ed.). American Technical Publishers, Inc.

Stephens, W.B. (Ed.) (1971) *Training the Developmentally Young*. New York: John Day.

Stern-Otazo, K.L. (1980). Curriculum modification and instruction practices. In Meers, G.D. (Ed.). *Handbook of Special Vocational Needs Education*. Rockville, MD: Aspen Systems Corporation.

Stern, S. & Gathercoal, F. (1986). Working with handicapped students. *School Shop*, January.

**Business Computer Programming I**  
**APPENDIX G: Special Needs**

Texas Education Agency (1989). *Office Education Handbook*. Austin, Texas. (Available from Occupational Curriculum Laboratory at East Texas State University, Commerce, Texas.)

Texas Education Agency (1985). *State Board of Education Rules for Curriculum*. Austin, Texas.

Tindall, L.W. (1980). *Puzzled about educating special need students? A handbook on modifying vocational curricula for handicapped students*. Wisconsin Vocational Studies Center. University of Wisconsin-Madison.



## 2 + 2 + 2 ARTICULATION

As a result of the changes in office information systems technology, the electronic office requires a work force with many diverse skills and attributes. The traditional job categories and job titles of office workers have been replaced with an encyclopedia of positions supporting office information systems functions and activities.

To train the office personnel required to support these positions, the two-year secondary office education training program or the two-year community college office careers degree program may not provide the extensive training that business and industry require. Most office information systems management positions require four-year college degrees.

In the past, office education programs offered through the high schools and the community colleges have presented duplicated training plans. Students who chose to enroll in community college office careers programs after high school often had to repeat many of the same competencies that they mastered in their high school vocational programs. If the students transferred to a four-year university program, many of the courses would not transfer. In most instances, the university program was a duplication of the high school and the community college training programs.

In order to prepare the office information systems work force of the future, high schools, community colleges, and universities must develop coordinated training programs that help students in making a smooth transition from one educational level to another without experiencing delays, duplication of effort, or loss of credit.

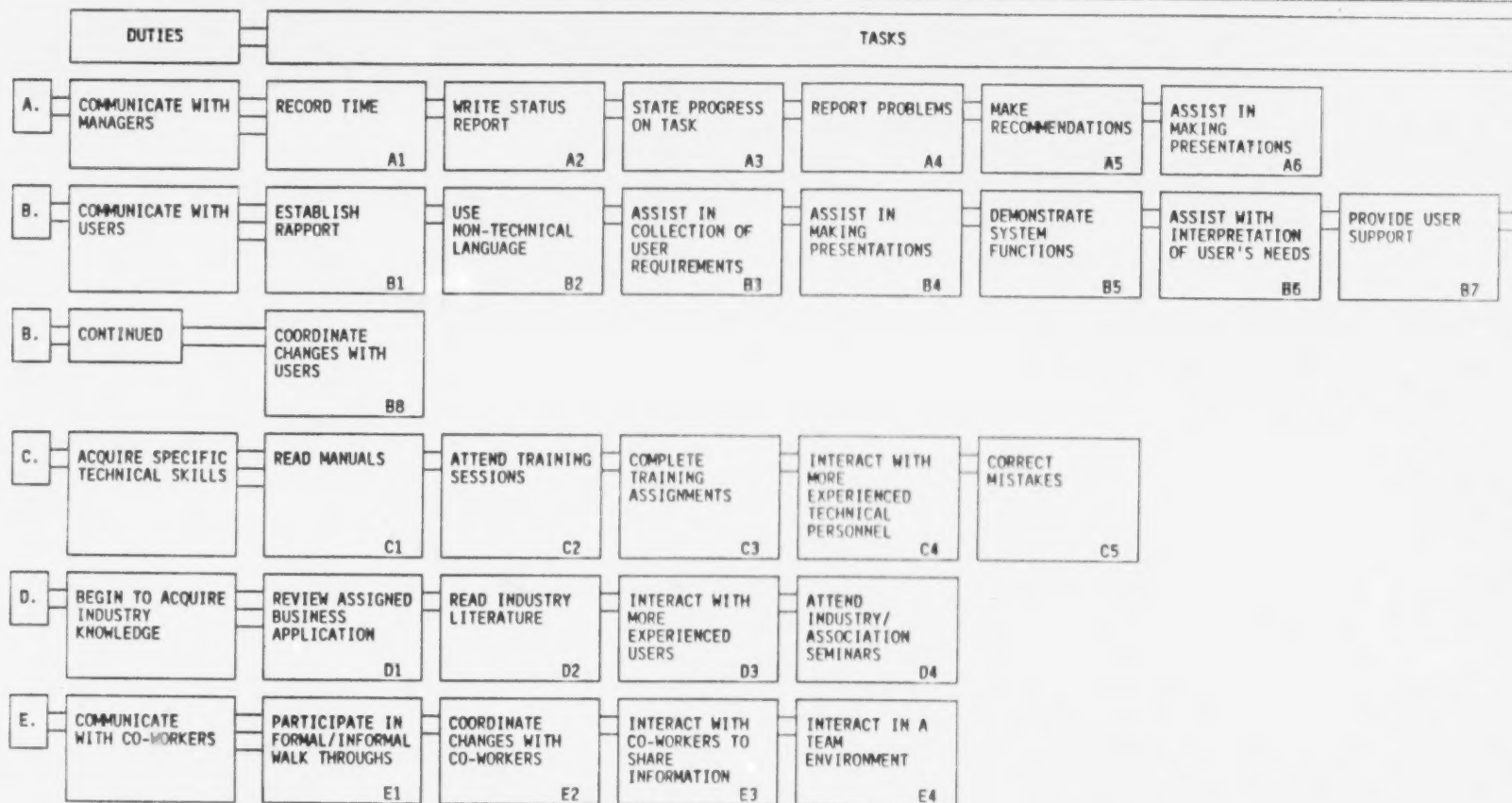
A 2 + 2 + 2 articulation program in the area of office information systems will be developed in this six-year project. The program will provide a complete six-year training program that begins in grade eleven and joins the two-year Dallas County Community College degree program with the junior and senior years of the four-year University of North Texas Office Technology program. Several other junior and senior colleges and universities in Texas have shown interest in participating in this articulation project. The 2 + 2 + 2 articulation program will be developed according to the following process:

- Phase 1: The project administrator and the curriculum development specialist will establish a steering committee to develop a plan of action for project implementation. The steering committee will be divided into subcommittees to provide input into the development of the surveys, competency chart, secondary curriculum guides, competency profile, and the evaluation, tracking, and follow-up system.
- Phase 2: Businesses and industries in the Dallas-Fort Worth area and in three additional metropolitan areas will be surveyed by the project administrator and the curriculum development specialist to determine employment needs and to define the occupational career path for office information systems from entry-level to management. Surveys will be completed and tabulated by November 15, 1987.

- Phase 3: DACUM analyses will be conducted of each major occupational category in the career path. The project administrator and the curriculum development specialist will serve as DACUM coordinators, and two trained DACUM consultants will be hired to facilitate the DACUMs. The first DACUM will be completed and validated by February 1, 1987. It is anticipated that the remaining DACUMS will be completed and validated by March 1, 1988. A competency continuum chart will then be developed by occupational category (entry-level to advanced).
- Phase 4: The project administrator and the curriculum development specialist will work with representatives from the Educational Development and Training Center (formerly Occupational Curriculum Laboratory), East Texas State University, Dallas County secondary and DCCCD office education (information systems) faculty, and representatives from business and industry to develop the curriculum guides (including course outlines and instructional materials) and competency profiles using the results of validated DACUM analyses. Curriculum guides will meet the approved TEA guidelines for the State office education occupational cluster curriculum. Curriculum guides and profiles will be completed by May 15, 1988.
- Phase 5: The project administrator and the curriculum development specialist will work with a subcommittee of Dallas County secondary and DCCCD office education (information systems) faculty to develop a process to evaluate and upgrade course content and curriculum materials. Evaluation system will be completed and tested by May 15, 1988.
- Phase 6: The project administrator and the curriculum development specialist will conduct a curriculum workshop for secondary office education (information systems) faculty in Dallas County to review the curriculum guides, including instructional materials and competency profiles. The workshop will be scheduled prior to June 15, 1988.
- Phase 7: The project administrator and the curriculum development specialist will work with a subcommittee of Dallas County secondary and DCCCD office education (information systems) faculty, occupational administrators, and placement personnel to develop a system to track students from grade eleven through the two-year community college and university programs. The Follow-Up System will be compatible with the new DCCCD CareerPlanning and Placement System. The Follow-Up System will be reviewed by the steering committee prior to June 15, 1988.
- Phase 8: The project administrator will work with vocational administrators from the Dallas Independent School Districts to revise existing articulation agreements. Agreements will be revised prior to June 30, 1988.
- NOTE: Phases 1 - 8 were completed according to the established schedule, and 2 + 2 + 2 articulation is presently an ongoing project.



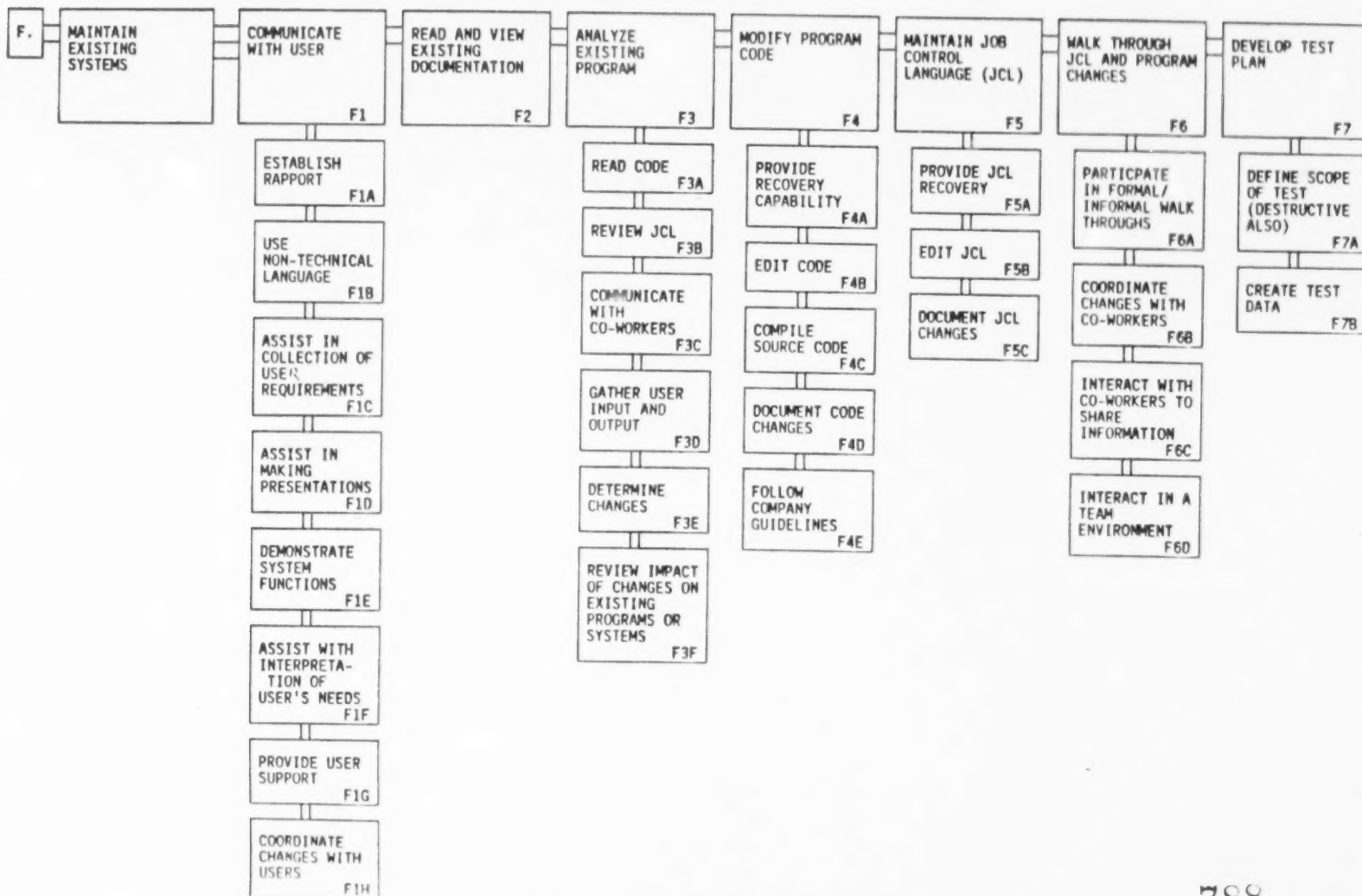
# COMPUTER PROGRAMMER/ANALYST DACUM CHART

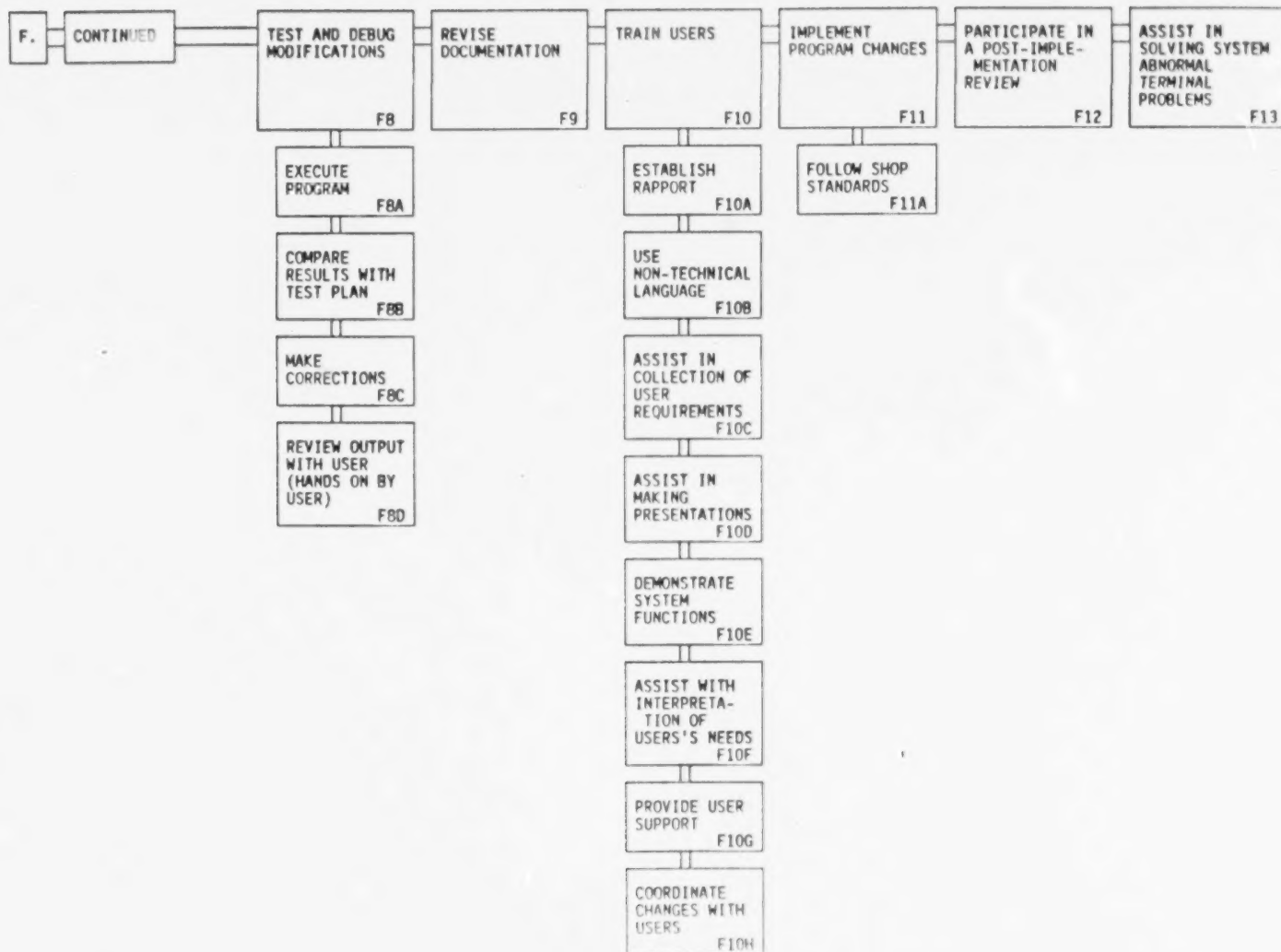


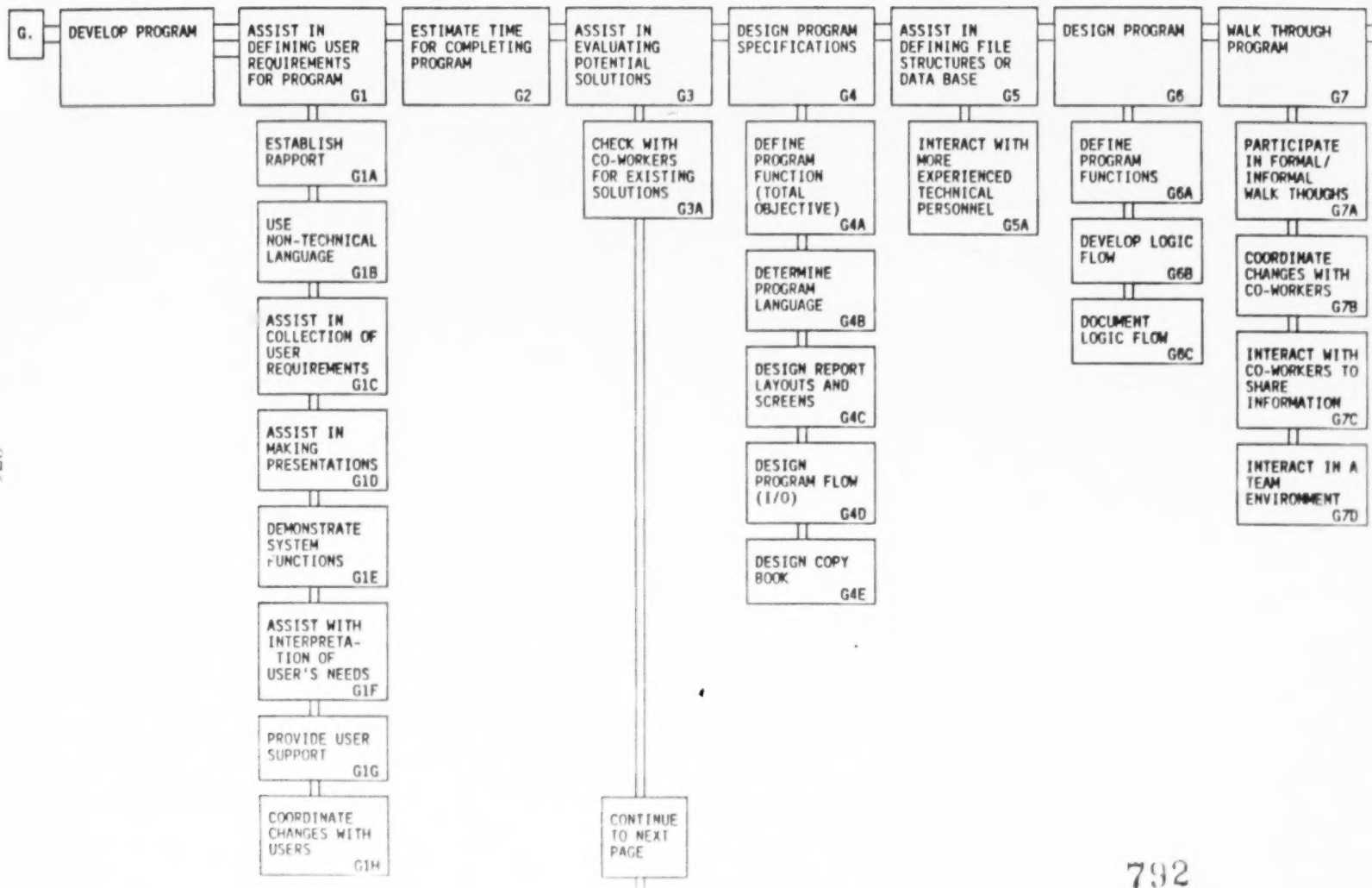
873

785

786

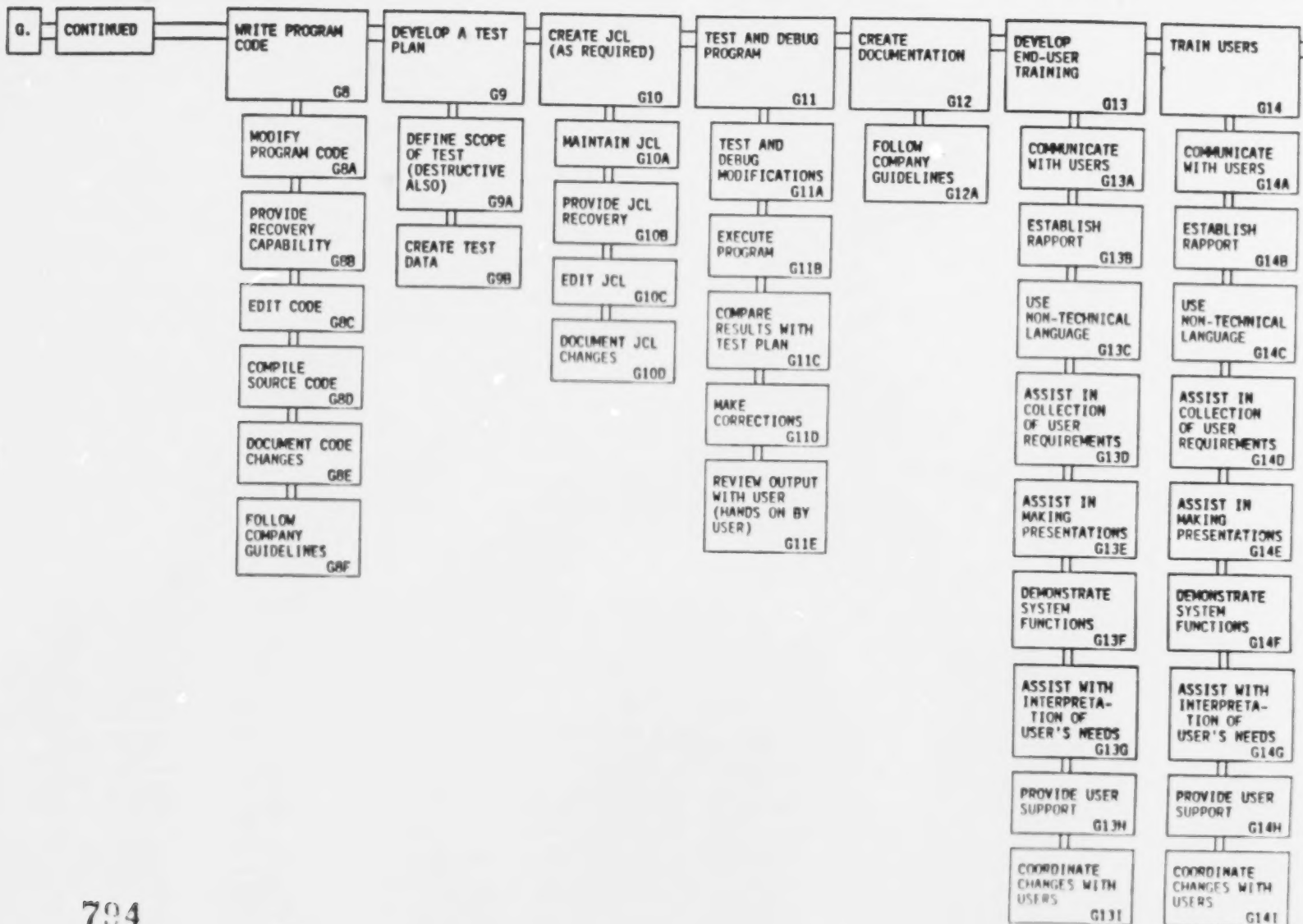


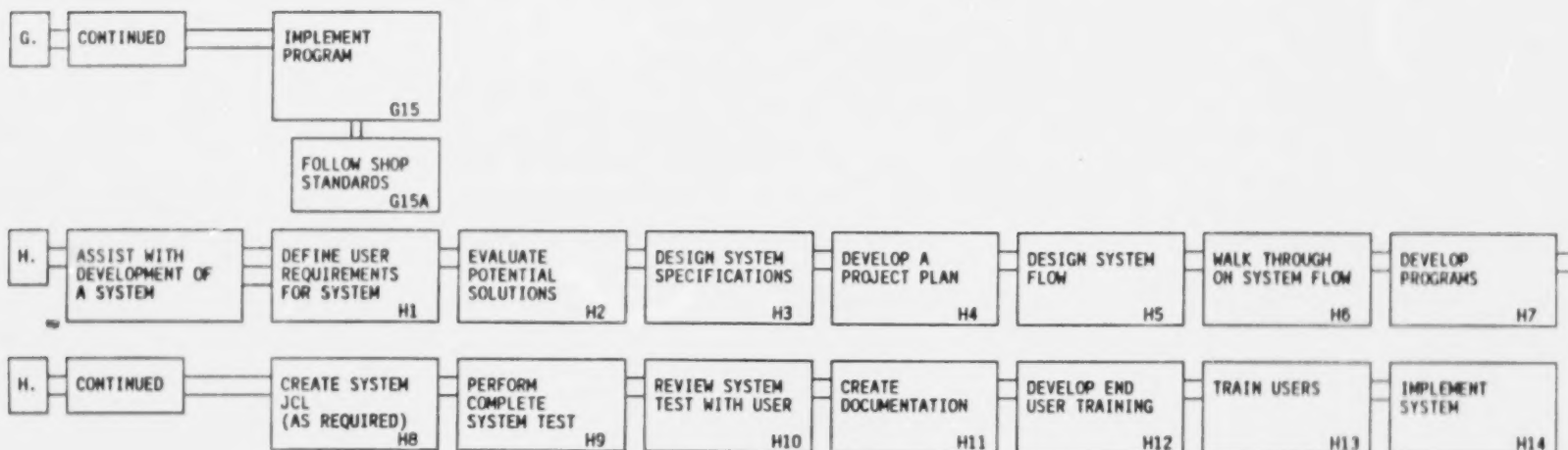




CONTINUED  
FROM PREV.  
PAGE







(First Job)  
PROGRAMMER/ANALYST

JOB TITLES

- JUNIOR PROGRAMMER/ANALYST
- SOFTWARE ENGINEER
- ANALYST/PROGRAMMER
- ASSOCIATE PROGRAMMER
- SYSTEM ENGINEER

ATTITUDES/TRAITS

- ADAPTABILITY
- WILLINGNESS TO CHANGE
- EXTROVERT PERSONALITY
- DETAIL-ORIENTED
- SELF-MOTIVATED - QUEST FOR KNOWLEDGE
- CORPORATE CITIZEN/VOLUNTEER EFFORT
- OK TO MAKE MISTAKES (ONCE)
- WILLINGNESS TO LEARN FROM MISTAKES
- WILLINGNESS TO ADMIT WHEN YOU NEED HELP
- WILLINGNESS TO SHARE/EXCHANGE INFORMATION WITH CO-WORKERS
- SUPPRESS EGO

KNOWLEDGE/SKILLS

SHOP STANDARDS

- NAMING
- CODING
- DOCUMENTATION
- JCL

- HOW TO MAKE ORAL AND WRITTEN PRESENTATIONS
- COMMUNICATION SKILLS (Including keyboarding and clerical skills)

• LANGUAGES  
PC

- BASIC
- "C"
- D-BASE-III
- PASCAL
- COBOL
- FORTRAN
- ASSEMBLY LANGUAGES

MAINFRAME/MINI COMPUTERS

- COBOL
- ASSEMBLY
- RPG (BIG ON SYSTEM MINIS AND CG's)
- FORTRAN/SCIENTIFIC
- FOURTH GENERATION LANGUAGE
  - FOCUS
  - RAMIS
  - SAS

- STRUCTURED PROGRAMMING TECHNIQUES
- CODE GENERATORS
- DATA PROCESS MANAGEMENT



- HIERARCHICAL/RELATIONAL PROGRAMMING
- ON-LINE PROGRAMMING
- "DUMP" READING
- ESTIMATING LENGTH OF TIME FOR PROJECT
- TESTS
  - DESTRUCTIVE WITH INVALID DATA
  - EDIT TESTING
  - OUT OF SEQUENCE/IN SEQUENCE NUMBER OF TRANSACTIONS
  - VOLUME
  - CROSS-TEST VARIETIES OF DATA
  - STRESS-TEST NUMBER OF PEOPLE ON-LINE SIMULTANEOUSLY
- DOCUMENTATION PROGRAMS
  - PSEUDO CODE
  - FLOW CHARTS
  - HIPO (HIERARCHICAL PROCESSING INPUT/PROCESSING OUTPUT)
  - V/TOC (VISUAL TABLE OF CONTENTS)
  - DECISION TABLES
  - WAINER/ORR
  - CHAPIN CHART/NASSI-SCHNEIDER MAP
- DATA BASES
  - IMS/CICS-DL1
  - ADR/DATA-COM
  - ADABAS
  - DB2 and SQL/DS
  - ORACLE (CROSSES MANY MACHINES)
  - INGRES
- TOOLS (SOFTWARE)
  - INTER-TEST )
  - TSO/ISPF )
  - LIBRARIAN ) THESE ARE ALL IBM TOOLS
  - PANVALET )
  - ROSCOE )
  - VOLLIE )

#### FUTURE PROJECTIONS

- EGOLESS PROGRAMMING
- PREPACKAGE APPLICATIONS i.e. "TURN-KEY"
- SHIFT FROM SKILLS IN PROGRAMMING TO RELATIONSHIPS WITH USER -- TO INTERPRET NEEDS AND TO ANALYZE AND DESIGN USER APPLICATIONS
- INCREASE IN INDUSTRY KNOWLEDGE COULD BECOME "USER" ANALYST (PARTICULARLY WITH FOURTH GENERATION LANGUAGES)

- DP IS AUTOMATING ITSELF
- FEWER TECHNICAL PEOPLE EMPLOYED TO DEVELOP SYSTEM/SOFTWARE PACKAGES; MOST WILL BE USER ANALYSTS
- PC PROLIFERATING MORE FUNCTIONS
  - LINKS TO MAINFRAME
  - MAINFRAME CAPABILITY MOVES TO DESKTOP
  - MAINFRAME MAINTAINS CENTRALIZED DATABASE
  - DATABASE/CENTRALIZED-DATA PROCESSING DECENTRALIZED

## **COMPLIANCE STATEMENT**

### **TITLE VI, CIVIL RIGHTS ACT OF 1964; THE MODIFIED COURT ORDER, CIVIL ACTION 5281, FEDERAL DISTRICT COURT, EASTERN DISTRICT OF TEXAS, TYLER DIVISION**

Reviews of local education agencies pertaining to compliance with Title VI Civil Rights Act of 1964 and with specific requirements of the Modified Court Order, Civil Action No. 5281, Federal District Court, Eastern District of Texas, Tyler Division are conducted periodically by staff representatives of the Texas Education Agency. These reviews cover at least the following policies and practices:

- (1) acceptance policies on student transfers from other school districts;
- (2) operation of school bus routes or runs on a non-segregated basis;
- (3) nondiscrimination in extracurricular activities and the use of school facilities;
- (4) nondiscriminatory practices in the hiring, assigning, promoting, paying, demoting, reassigning, or dismissing of faculty and staff members who work with children;
- (5) enrollment and assignment of students without discrimination on the basis of race, color, or national origin;
- (6) nondiscriminatory practices relating to the use of a student's first language; and
- (7) evidence of published procedures for hearing complaints and grievances.

In addition to conducting reviews, the Texas Education Agency staff representatives check complaints of discrimination made by a citizen or citizens residing in a school district where it is alleged discriminatory practices have occurred or are occurring.

Where a violation of Title VI of the Civil Rights Act is found, the findings are reported to the Office for Civil Rights, Department of Health, Education and Welfare.

If there is a direct violation of the Court Order in Civil Action No. 5281 that cannot be cleared through negotiation, the sanctions required by the Court Order are applied.

### **TITLE VII, CIVIL RIGHTS ACT OF 1964; EXECUTIVE ORDERS 11246 AND 11375; TITLE IX, 1973 EDUCATION AMENDMENTS; REHABILITATION ACT OF 1973 AS AMENDED; 1974 AMENDMENTS TO THE WAGE-HOUR LAW EXPANDING THE AGE DISCRIMINATION IN EMPLOYMENT ACT OF 1967; AND VIETNAM ERA VETERANS READJUSTMENT ASSISTANCE ACT OF 1972 AS AMENDED IN 1974.**

It is the policy of the Texas Education Agency to comply fully with the nondiscrimination provisions of all federal and state laws and regulations by assuring that no person shall be excluded from consideration for recruitment, selection, appointment, training, promotion, retention, or any other personnel action, or be denied any benefits or participation in any programs or activities which it operates on the grounds of race, religion, color, national origin, sex, handicap, age, or veteran status (except where age, sex, or handicap constitute a bona fide occupational qualification necessary to proper and efficient administration). The Texas Education Agency makes positive efforts to employ and advance in employment all protected groups.